


МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего
образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

На правах рукописи



Мошкин Вадим Сергеевич

**ИССЛЕДОВАНИЕ ПРЕДСТАВЛЕНИЯ
ТЕРМИНОЛОГИИ В ЛИНГВИСТИЧЕСКОМ
ОБЕСПЕЧЕНИИ САПР НА ОСНОВЕ ИНТЕГРАЦИИ
НЕЧЕТКИХ ОНТОЛОГИЙ И ЛОГИЧЕСКОГО
ВЫВОДА**

Специальность 05.13.12 – Системы автоматизации проектирования
(промышленность)

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель – доктор технических наук, профессор
Ярушкина Надежда Глебовна

Ульяновск – 2017

Оглавление

Введение	6
Глава 1 Анализ современного состояния методов и систем в интеллектуальных САПР	19
1.1 Проектирование сложных технических систем.....	19
1.2 Лингвистическое обеспечение САПР	21
1.3 Современные тенденции интеллектуализации САПР	22
1.4 Экспертные системы, как элемент интеллектуальной САПР	24
1.4.1 Средства представления терминологии в интеллектуальных САПР	26
1.4.2 Логический вывод в интеллектуальных САПР	28
1.4.3 Нечеткость в логическом выводе знаний	28
1.4.4 Анализ существующих систем, реализующих алгоритм НЛВ	32
1.5 Нечеткие онтологии.....	36
1.5.1 Методика построения нечеткой онтологии из разных источников.....	37
1.5.2 Расширение OWL с помощью FuzzyOWL	39
1.5.3 Свойства-аннотации OWL	39
1.5.4 Формальная модель нечеткой онтологии FuzzyOWL	40
1.6 Методики извлечения терминов в задачах расширения ядра онтологии ..	43
1.7 Постановка задачи исследования	52
Глава 2 Методы и алгоритмы интеграции онтологического анализа и логического вывода	55
2.1 Выбор класса модели онтологии для САПР	55
2.2 Формальная модель онтологии ПрО.....	57
2.3 Алгоритм интеграции онтологии и логического вывода	59
2.4 Использование прецедентов в процессе логического вывода	66
2.5 Формальная модель системы вывода, основанной на анализе прецедентов	68
2.6 Алгоритмы расширения онтологии	71
2.6.1 Тезаурусный алгоритм	72

2.6.2 Алгоритм вложенных связей	76
2.7 Выводы по главе	79
Глава 3 Реализация интеллектуальной программной системы логического вывода на основе онтологии для интеллектуальной САПР	80
3.1 Выбор инструментов инженерии онтологии объекта проектирования	80
3.2 Диаграмма вариантов использования разработанной системы	83
3.3 Подсистема онтологически-ориентированного логического вывода на основе нечеткой онтологии.....	85
3.3.1 Описание проекта интеллектуальной компоненты САПР	86
3.3.2 Диаграмма классов	87
3.3.3 Реализация алгоритма интеграции.....	88
3.4 Подсистема автоматизированного расширения ядра онтологии.....	94
3.4.1 Базовый функционал системы.....	94
3.4.2 Реализация базы данных	95
3.4.3 Схема информационного обеспечения.....	96
3.4.4 Диаграмма состояний	96
3.4.5 Схема работы.....	99
3.5 Выводы по главе	102
Глава 4 Анализ адекватности разработанных моделей и методов на основе вычислительных экспериментов и практики применения	104
4.1 План проведения экспериментов	104
4.2 Эксперименты по интеграции онтологического анализа и логического вывода в задачах автоматизированного проектирования.....	106
4.2.1 Онтология ЛВС	107
4.3 Сравнение методик логического вывода на основе четких и нечетких онтологий	111
4.4 Проведение экспериментов по логическому выводу с элементами интерактивного обучения.....	112
4.5 Анализ эффективности семантической оценки терминологичности.....	114
4.5.1 Онтология «Станки с ЧПУ».....	115

4.5.2 Метрики оценки результатов.....	117
4.5.3 Результаты вычислительных экспериментов извлечения терминов из проектной документации	118
4.5.4 Использование результатов работы предложенных алгоритмов при проведении отраслевой подготовки специалистов по мультязычной проектной документации	125
4.6 Применение АС поддержки проектирования в деятельности АО «Авиастар-СП».....	133
4.6.1 Характеристики объекта автоматизации.....	135
4.6.2 Направления применения разработанной системы.....	138
4.7 Выводы по главе	141
Заключение	143
Библиографический список	145
Приложение А. Свидетельства о государственной регистрации программ для ЭВМ.....	161
Приложение Б. Акт внедрения	164
Приложение В. Диплом победителя конкурса УМНИК.....	166
Приложение Г. Таблица вывода рекомендаций при проведении экспериментов по оценке состояния ЛВС	167
Приложение Д. Пример представления нечеткого правила в формате SWRL	174
Приложение Е. Фрагмент FuzzyOWL-онтологии оценки состояния ЛВС....	180
Приложение Ж. Код класса работы с нечёткими объектами OWL-онтологии	186

Список сокращений

OWL – Web Ontology Language

RDF(S) – Resource Description Framework (Schema).

SCADA – Supervisory Control And Data Acquisition.

SWRL – Semantic Web Rule Language

АП – автоматизированное проектирование.

АСУ – автоматизированная система управления.

БД – база данных.

БЗ – база знаний.

ВР – временной ряд.

ДЛ – дескрипционная логика.

ИИ – искусственный интеллект.

ЛО – лингвистическое обеспечение.

ЛВС – локальная вычислительная сеть.

НЛВ – нечеткий логический вывод.

ПЛС – поточная линия сборки.

ПО – программное обеспечение.

ПрО – предметная область.

САПР – система автоматизированного проектирования.

СТС – сложная техническая система.

СУБД – система управления базами данных.

ТП – технологический процесс.

ЧПУ – числовое программное управление.

Введение

Проектирование технических систем представляет собой сложный процесс, основанный на последовательности принятия проектных решений, нуждающихся в инструментальных средствах, а также в лингвистическом обеспечении, включающем в себя предметную терминологию. В связи с этим, системы автоматизированного проектирования (САПР), предназначенные для решения проектных задач, должны использовать модели, алгоритмы и методы манипулирования знаниями, позволяющие учитывать особенности объектов предметной области (ПО) и их отношений и использовать их с целью повышения качества принимаемых проектировщиком решений.

Современные САПР обладают развитым лингвистическим обеспечением, традиционной частью которого являются тезаурусы, предметные словари и онтологии. Лингвистическое обеспечение САПР включает в себя совокупность языков, используемых при автоматизированном проектировании, терминов и определений, правил формализации естественного языка, а также методов сжатия и развертывания текстов (ГОСТ 22487-77) [76].

В настоящее время большинство интеллектуальных САПР используют для хранения знаний продукционную модель, позволяющую делать логический вывод рекомендаций с помощью заранее сформированного экспертом набора правил. Недостатком данного метода является неполнота описания предметной области, помимо этого, модель требует строгой структуризации экспертных знаний, что в условиях слабой формализованности данных о ПрО делает использование данной модели оправданным лишь в ограниченных количествах случаев.

Декларативные модели представления экспертных знаний в контексте анализа проектных данных позволяют связать результаты анализа с рассматриваемой предметной областью, но не дают возможности делать логический вывод рекомендаций по результатам анализа, вследствие отсутствия встроенного логического вывода в основе моделей.

Использование одного формата представления знаний существенно снижает гибкость и функциональность интеллектуальных САПР, используемых в процессе проектирования сложных технических систем, именно поэтому в настоящее время актуальной является разработка методов и алгоритмов интеграции разных моделей представления слабоформализованных экспертных знаний. Наиболее универсальной и полной с точки зрения охвата специфики предметной области во всем разнообразии отношений между ее объектами является модель интеграции онтологического и продукционного подходов представления знаний, позволяющая в процессе логического вывода соответствующих рекомендаций опираться на данные, представленные в виде онтологии и наиболее полно описывающие рассматриваемую предметную область.

Одним из направлений использования онтологических структур в САПР является индексирование технической документации на основе онтологий, содержащих ограниченный набор отношений между категориями – тезаурусов. Каждому объекту (категории) тезауруса приписывается некоторый пояснительный текст, таким образом регламентируя между ними отношения подчиненности, синонимии, иерархии и т.д. Структура подобных тезаурусов регламентируется следующими стандартами:

- ISO 2788:1986 Документация. Руководство по построению и разработке одноязычных тезаурусов [15];
- ГОСТ 7.25-80 [75] - для одноязычных тезаурусов;
- ГОСТ 7.24-2007 [74] - для многоязычных тезаурусов.

В процессе проектирования сложных технических систем (СТС) возникает потребность в унификации используемой терминологии с целью обеспечения связности процесса производства. Эта задача усложняется участием в процессе специалистов разных компетенций и уровней подготовки, а также различными форматами используемой документации.

Таким образом, терминология предметной области в составе лингвистического обеспечения САПР имеет большое значение в процессе

проектирования, т.к. она обеспечивает проект словарной базой и является основной для проектной и сопутствующей документации. Здесь возникает задача извлечения терминологии из текстов по предметной области и ее представления. Представление терминологии имеет целью создание аппарата терминов и понятий, формирующих лингвистическое обеспечение САПР, в виде словарей, тезаурусов и онтологий.

В диссертационной работе предлагается подход к построению интеллектуальной компоненты САПР, создающей единый лингвистический и информационный базис процесса принятия проектных решений на экспертном уровне и основанной на интеграции и гибридизации нечетких моделей, алгоритмов онтологического анализа и механизма прецедентов.

Существенных результатов в области разработки интеллектуальных САПР с использованием онтологической формы представления предметной терминологии добились С.В. Смирнов (разработка среды моделирования инженерных знаний, концептуальное моделирование) [121, 122, 123] Н.М. Боргест (автоматизированное проектирование в области самолетостроения, разработка работа-проектанта)[41, 60, 59], В.А. Виттих (разработка методов и средств создания открытых мультиагентных систем для поддержки процессов принятия проектных решений) [45], П.И. Соснин (вопросно-ответные рассуждения в среде автоматизированного проектирования, псевдо-кодированное программное управление в проектировании) [125, 28, 27], В.М. Курейчик (генетические алгоритмы в интеллектуальных САПР, эволюционное проектирование) [92, 81, 57] и др.

Начало разработки интеллектуальных программных систем, основанных на онтологическом подходе, связано с зарубежными исследователями, среди которых Т. Gruber, N. Guarino, M. Uschold, F. Bobillo, U. Straccia, G. Rzevski и т.д. Среди отечественных ученых проблемы использования онтологий и аналогичных им семантических моделей описания ПрО и поддержки принятия решений в сфере автоматизированного проектирования исследуют Ю.А. Загорулько, Т.А. Гаврилова, Н.Г. Загоруйко, В.Ф. Хорошевский, А.С. Клещев,

В.В. Грибова, Д.А. Новиков, В.В. Голенков, В.А. Виттих, А.В. Смирнов, С.В. Смирнов, А.Ф. Тузовский и др.

Однако, в настоящее время постоянный рост требований к эффективности и качеству лингвистического и информационного обеспечения автоматизированного проектирования технических систем в условиях слабой формализации поставленных задач, особенно на ранних этапах проектирования, предполагает необходимость системного решения ряда научных задач:

- необходимость унификации процесса автоматизированной обработки терминологии в лингвистическом обеспечении при решении проектных задач различного рода;
- необходимость разработки семантического базиса анализа представления терминологии в рамках решения задач автоматизированного проектирования;
- отсутствие интегративных концептуальных моделей, использующих различные подходы представления знаний о предметной области в форме тезауруса;
- необходимость одновременного использования разноаспектных описаний особенностей рассматриваемой предметной области;
- необходимость решения проблемы учета опыта решения схожих проектных задач;
- необходимость адаптации и расширения лингвистического и информационного обеспечения проектной деятельности вследствие изменения особенностей ПрО;
- необходимость решения проблемы учета нечеткости в рассуждениях проектировщика.

Поэтому актуальной является тема диссертации, посвященная разработке методов и средств представления терминологии в лингвистическом обеспечении САПР на основе интеграции нечетких онтологий и логического вывода в задачах автоматизированного проектирования.

Цель диссертационной работы

Целью диссертации является разработка и реализация эффективных моделей и алгоритмов представления терминологии в лингвистическом обеспечении САПР на основе интеграции нечетких онтологий и логического вывода, обеспечивающих снижение количества ошибок проектировщика.

Предмет исследования

Модели, методы и средства представления терминологии в лингвистическом обеспечении САПР на основе интеграции нечетких онтологий и логического вывода в процессе автоматизированного проектирования.

Объект исследования

Объектом исследования является лингвистическое обеспечение автоматизированного проектирования технических систем. В качестве примера исследованы процессы проектирования автоматизированной системы поточной линии сборки самолета, включающие ЛВС, линии станков с ЧПУ.

Задачи исследования

В соответствии с целью работы актуальными являются следующие задачи диссертационного исследования:

- провести сравнительный анализ современных средств представления терминологии в лингвистическом обеспечении интеллектуальных САПР;
- разработать онтологически-ориентированную систему извлечения терминологии из проектных документов с использованием сформированного ядра предметной онтологии;
- провести сравнительный анализ современных интеллектуальных методов логического вывода в САПР, выявить их возможности и ограничения в плане решения задач автоматизированного проектирования;
- рассмотреть возможность применения методов онтологического анализа для решения задач оценки состояния сложного технического объекта рассматриваемой предметной области с целью поддержки принятия проектных

решений на экспертном уровне;

- исследовать возможность интеграции методов теории нечетких систем и онтологического анализа состояния сложной технической системы;
- разработать онтологическую модель представления набора продукций и алгоритм логического вывода рекомендаций на экспертном уровне в задачах автоматизированного проектирования;
- разработать алгоритм расширения процедурной составляющей базы знаний интеллектуальной САПР с использованием механизма прецедентов;
- разработать алгоритм расширения декларативного описания ПрО с использованием сформированного ядра онтологии интеллектуальной САПР посредством извлечения терминологии из проектных документов;
- разработать автоматизированную систему (АС) поддержки проектирования, реализующую алгоритм интеграции механизмов логического вывода и онтологического представления экспертных знаний с элементами нечеткости, а также использующей механизм учета прецедентов с целью расширения ядра онтологии;
- провести вычислительные эксперименты, позволяющие оценить эффективность предложенных моделей и алгоритмов в процессе проведения концептуального проектирования сложных технических систем;
- внедрить результаты исследований в практику процесса проектирования технических систем предприятий региона.

Нужно отметить, что при решении задачи оценки эффективности предложенных моделей и алгоритмов необходима адаптация условий проведения экспериментов под специфику решаемых задач.

Методы исследования

В диссертационной работе применяются методы онтологического анализа, дескрипционной логики, теории нечетких систем, а также объектно-ориентированного программирования при построении программного комплекса.

Научная новизна

Научная новизна результатов исследования заключается в следующем:

1. Предложена онтологически-ориентированная методика извлечения терминологии из проектных документов, отличающаяся от известных совместным использованием статистических методов, лингвистических шаблонов и базового ядра предметной онтологии;
2. Разработан алгоритм расширения онтологического представления терминологии предметной области технической системы, отличающийся от известных построением вложенных связей термина-кандидата, извлеченного из проектного документа, до опорного класса базового ядра онтологии;
3. Разработан алгоритм расширения процедурной составляющей базы знаний интеллектуальной САПР, отличающийся от известных использованием в качестве прецедента совокупности SWRL-правил и правил решения задачи проектирования;
4. Предложен метод интеграции нечеткого логического вывода и нечеткого онтологического представления терминологии предметной области технической системы, отличающийся от известных использованием иерархического нечеткого вывода и модели онтологии класса FuzzyOWL;
5. Разработана программная система поддержки проектирования, отличающаяся наличием специальной компоненты интеллектуальной САПР для представления терминологии лингвистического обеспечения.

Практическая значимость работы

Разработанная интеллектуальная компонента САПР, реализующая алгоритмы интеграции онтологического анализа и логического вывода на основе нечетких онтологий с возможностью расширения базы знаний терминологии посредством гибридизации механизма прецедентов и алгоритмов извлечения терминологии из проектных документов, была использована в рамках проекта разработки проектной документации Автоматизированной системы управления (АСУ) поточной линии сборки АО «Авиастар-СП». Также данная компонента используется при построении универсальной базы знаний учебного центра

«Юнитех» агентства переводов «Юнитранс» (Приложение Б).

Соответствие паспорту специальности.

Согласно паспорту специальности 05.13.12 «Системы автоматизации проектирования (промышленность)», проблематика, рассмотренная в диссертации, соответствует следующим областям исследований: «Разработка научных основ построения средств САПР, разработка и исследование моделей, алгоритмов и методов для синтеза и анализа проектных решений, включая конструкторские и технологические решения в САПР и АСТПП».

Основания для выполнения работы

Результаты диссертационной работы использовались в ряде НИОКР, выполненных в Ульяновском государственном техническом университете, направленных на решение научно-технических задач. К наиболее важным результатам следует отнести:

1. Участие в выполнении гранта РФФИ №13-01-00324 «Исследование формальных методов грануляции слабоструктурированных информационных ресурсов на основе онтологии предметной области».
2. Участие в выполнении гранта РФФИ №15-41-02413 «Интеллектуальный анализ временных рядов на основе нечетких онтологий, извлекаемых из Интернет-ресурсов».
3. Участие в выполнении гранта РФФИ № 16-47-730742 «Интеграция онтологических моделей и проектных диаграмм при концептуальном проектировании сложных информационных систем».
4. Участие в реализации гранта по программе УМНИК в области информационных технологий при поддержке Фонда содействия развитию малых форм предприятий в научно-технической сфере.
5. Участие в реализации государственного задания №2014/232 на выполнение государственных работ в сфере научной деятельности Минобрнауки России по проекту «Разработка нового подхода к интеллектуальному анализу слабоструктурированных информационных ресурсов».

Достоверность результатов диссертационной работы

Достоверность научных положений, выводов и рекомендаций подтверждена результатами вычислительных экспериментов, а также результатами использования материалов диссертации в работе компаний «Юнитранс» и АО «Авиастар – СП».

Основные положения, выносимые на защиту

1. Онтологически-ориентированная методика извлечения терминологии из проектных документов с использованием базового ядра предметной онтологии;
2. Алгоритм расширения онтологического представления терминологии предметной области технической системы с использованием базового ядра онтологии интеллектуальной САПР посредством извлечения терминологии из проектных документов;
3. Алгоритм расширения процедурной составляющей базы знаний интеллектуальной САПР с использованием механизма прецедентов;
4. Метод интеграции нечеткого логического вывода и нечеткого онтологического представления терминологии предметной области технической системы;
5. Программная система поддержки проектирования, являющаяся компонентой интеллектуальной САПР для представления терминологии лингвистического обеспечения.

Апробация работы

Основные положения и результаты диссертационной работы докладывались, обсуждались и получили одобрение на следующих конференциях, семинарах и симпозиумах: V-ой всероссийской научно-технической конференции аспирантов, студентов и молодых ученых «ИВТ-2013» (г. Ульяновск, 2013 г.); 47-ой научно-технической конференции УлГТУ (г. Ульяновск, 2013 г.); VII-ой международной научно-практической конференции «Интегрированные модели и мягкие вычисления в искусственном интеллекте» (г. Коломна, 2013 г.); 48-ой научно-технической конференции УлГТУ (г. Ульяновск, 2014 г.); I-ой международной

Поспеловской летней школе-семинаре для студентов, магистрантов и аспирантов «Методы и технологии гибридного и синергетического искусственного интеллекта» (г.Светлогорск, 2014 г.); VI-ой всероссийской научно-технической конференции аспирантов, студентов и молодых ученых «ИВТ-2014» (г. Ульяновск, 2014 г.); II-м международном Поспеловском симпозиуме «Гибридные и синергетические интеллектуальные системы» (г. Светлогорск, 2014 г.); VI-ой Всероссийской научно-практической конференции «Нечеткие системы и мягкие вычисления» (г. Санкт-Петербург, 2014 г.); XIV-ой национальной конференции по искусственному интеллекту с международным участием «КИИ-2014» (г. Казань, 2014 г.); 49-ой научно-технической конференции УлГТУ (г. Ульяновск, 2015 г.); VII-ой всероссийской научно-технической конференции аспирантов, студентов и молодых ученых «ИВТ-2015» (г. Ульяновск, 2015 г.); V-ой международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (г. Минск, 2015 г.); VIII-ой международной научно-практической конференции «Интегрированные модели и мягкие вычисления в искусственном интеллекте» (г. Коломна, 2015 г.); VI-ой международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (г. Минск, 2016 г.), I-ой Международной научной конференции «Интеллектуальные информационные технологии в технике и на производстве» (г. Сочи, 2016 г.), XV-ой национальной конференции по искусственному интеллекту с международным участием «КИИ-2016» (г. Смоленск, 2016 г.), VII-ой международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (г. Минск, 2017 г.).

Научные публикации

По результатам работы было опубликовано 35 статей, 10 из которых в журналах из перечня ВАК, а также 1 статья в издании, индексируемом в Scopus. Получены свидетельства о государственной регистрации программ для ЭВМ № 2014616701 (2014 г.) «Онтологически-ориентированная система логического вывода», № 2014616248 (2014 г.) «Онтологически-ориентированная система

извлечения терминологии», № 2017611364 (2017 г.) «Интеллектуальная экспертная система на основе нечеткой онтологии».

Структура и объем диссертации

Диссертация состоит из введения, четырех глав, заключения, списка использованной литературы и приложений. Основное содержание работы изложено на 190 страницах, включая 44 рисунка и 15 таблиц. Список использованных источников состоит из 141 наименования.

В **первой главе** приводится описание современных тенденций интеллектуализации САПР, определение и структура лингвистического обеспечения САПР. Помимо этого, рассмотрена архитектура экспертных систем как элемента интеллектуальных САПР, и приведены основные средства и алгоритмы представления предметной терминологии в лингвистическом обеспечении САПР.

Выявлена необходимость введения нечеткости в процесс логического вывода, приведены основные алгоритмы нечеткого логического вывода, а также проведен анализ современных САПР, реализующих рассмотренные алгоритмы. В качестве отдельного класса представления предметной терминологии рассмотрен класс нечетких онтологий. Помимо этого, приведена методика разработки нечеткой онтологии, формируемой из различных источников, а также рассмотрена методика представления нечетких онтологий FuzzyOWL. Также приведены основные статистические и лингвистические алгоритмы извлечения терминологии из текстовых, в том числе и проектных, документов при решении задачи расширения ядра онтологии интеллектуальной САПР.

Во **второй главе** приводится описание модели предметной онтологии, используемой в качестве формы представления предметной терминологии в лингвистическом обеспечении САПР в процессе проектирования сложных технических систем.

Также проведена формализация онтологии предметной области, содержащей элементы нечеткости. Помимо этого, приведено описание разработанного алгоритма интеграции механизмов логического вывода и знаний

как в четкой форме, так и в форме онтологии, содержащей элементы нечеткости.

Описан вариант модификации алгоритмов вывода посредством использования механизма учета прецедентов. Представлено описание онтологически-ориентированной методики извлечения терминологии из проектных документов, включающую в себя два алгоритма расширения онтологического представления терминологии предметной области технической системы с использованием базового ядра онтологии интеллектуальной САПР посредством извлечения терминологии из проектных документов.

В **третьей главе** проведен анализ современных инструментов разработки онтологий, а также машин вывода, необходимых для реализации логического вывода.

Рассмотрена архитектура и функциональные возможности разработанной программной системы поддержки проектирования, являющейся компонентой интеллектуальной САПР для представления терминологии лингвистического обеспечения, реализующей алгоритм интеграции механизмов логического вывода и онтологического представления экспертных знаний с элементами нечеткости, а также использующей механизм учета прецедентов с целью расширения ядра онтологии.

В **четвертой главе** представлены результаты экспериментов по применению предложенных алгоритмов, реализованных в программной системе поддержки проектирования, на базе процесса сборки самолетов для производства тяжелого военно-транспортного самолета Ил-76МД- 90А на АО «Авиастар-СП».

В рамках исследования были проведены эксперименты по анализу состояния ЛВС в условиях искусственного повышения трафика с использованием разработанной интеллектуальной компоненты САПР, реализующей предложенные модели и алгоритмы интеграции и расширения экспертной базы знаний. Также проведен сравнительный анализ результатов экспериментов с использованием в качестве базы знаний АС поддержки

принятия проектных решений четких и нечетких онтологий. Помимо этого, приведены краткое описание и фрагмент используемой предметной онтологии оценки состояния ЛВС.

Представлены результаты экспериментов по извлечению терминологии из проектной документации на основании разработанных алгоритмов при построении универсальной базы знаний учебного центра «Юнитех» агентства переводов «Юнитранс». Приведены краткое описание и фрагмент используемой предметной онтологии «Станки с ЧПУ», а также проведена оценка эффективности полученных результатов.

Также представлены эксперименты по применению предложенных алгоритмов представления и извлечения предметной терминологии в процессе разработки проектной документации для Автоматизированной системы управления Поточной линией сборки АО «Авиастар-СП».

В приложение входят:

- А. Свидетельства о государственной регистрации программ для ЭВМ.
- Б. Акт внедрения результатов диссертационного исследования.
- В. Диплом победителя конкурса УМНИК в области информационных технологий при поддержке Фонда содействия развитию малых форм предприятий в научно-технической сфере.
- Г. Таблица вывода рекомендаций при проведении экспериментов по оценке состояния ЛВС.
- Д. Пример представления нечеткого правила в формате SWRL.
- Е. Фрагмент FuzzyOWL – онтологии оценки состояния ЛВС.
- Ж. Код класса работы с нечеткими объектами OWL-онтологии.

Личный вклад

Все результаты, составляющие содержание диссертации, получены автором самостоятельно. Подготовка к публикации некоторых результатов проводилась совместно с соавторами, причем вклад диссертанта был определяющим.

Глава 1 Анализ современного состояния методов и систем в интеллектуальных САПР

1.1 Проектирование сложных технических систем

Понятие «сложной системы», согласно [66], предполагает наличие большого количества взаимодействующих структурных элементов, посредством которых сама система приобретает новые свойства. С другой стороны, системы, состоящие из большого числа однородных элементов, выполняющих схожие функции, нельзя называть сложными.

Отсюда при разделении систем на сложные и простые необходимо учитывать возможность оценки сложности по структуре и функционированию: чаще всего встречаются системы, сложность которых состоит в сочетании и чередовании функций, намного реже – сложные по структуре, но относительно простые по функционированию.

Еще более узкий класс – сложные технические системы (СТС). Любая СТС состоит из двух подсистем - исполнительной и управляющей: первая реализует главную функцию системы, вторая обеспечивает управление поведением системы в зависимости от динамически изменяющихся условий среды.

СТС предполагают наличие следующих характеристик:

- иерархичность и модульность построения;
- быстродействие;
- возможность масштабирования;
- возможность реализации как основного, так и вспомогательного функционала;
- наличие унифицированных интерфейсов взаимодействия с другими системами;
- адаптивность;

- надежность и безопасность функционирования;
- возможность модернизации;
- удобство управления;
- стандартизация и т.д. [96]

Для СТС и для их элементов, реализующих новые технологические возможности, зачастую в каждом конкретном случае приходится заново выполнять многоуровневое иерархическое проектирование.

В связи с этим, помимо необходимости унификации получаемых изделий, встает вопрос об унификации проектных процедур в рамках систем автоматизированного проектирования (САПР), посредством создания единого информационного пространства процесса проектирования. На рис.1.1 представлен фрагмент схемы СТС производственного типа на примере поточной линии сборки самолетов.

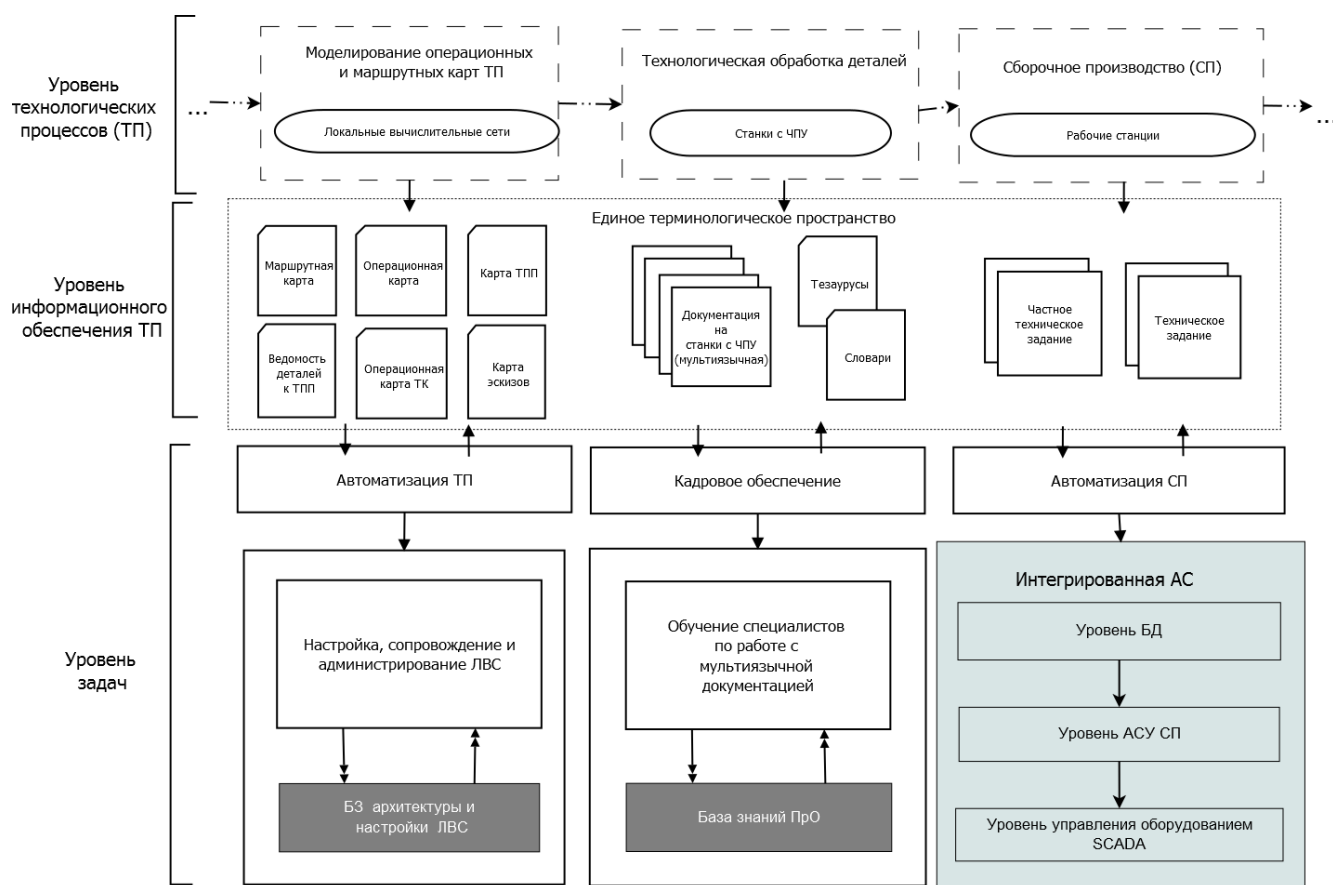


Рисунок 1.1 Фрагмент схемы СТС производственного типа на примере поточной линии сборки самолетов

Как видно из схемы, процесс проектирования СТС предполагает

необходимость наличия единого унифицированного терминологического базиса на уровне лингвистического обеспечения. Наличие подобного базиса позволяет избежать многочисленных коллизий и ошибок в процессе проектирования СТС, тем самым повысив его эффективность.

1.2 Лингвистическое обеспечение САПР

Согласно ГОСТ 22487-77 [76], «лингвистическое обеспечение САПР (ЛО САПР) представляет собой совокупность языков проектирования, включая термины и определения, правила формализации естественного языка, методы сжатия и развертывания текстов, необходимых для выполнения автоматизированного проектирования».

С одной стороны, с помощью ЛО САПР осуществляется непосредственное взаимодействие проектировщика с программными средствами поддержки проектирования, с другой - ЛО объединяет совокупность языковых средств для описания объектов проектирования, позволяя создавать единый терминологический базис, на основе которого разворачиваются все информационные потоки, циркулирующие между субъектами проектирования.

ЛО САПР образуют следующие языки:

- языки управления. Они служат для формирования соответствующих команд управления различным технологическим оборудованием, устройствами документирования и другими вспомогательными устройствами, используемыми в процессе автоматизированного проектирования;
- языки программирования. Они необходимы непосредственно для разработки САПР;
- языки проектирования. Данный набор языков ориентирован в первую очередь на проектировщиков и предназначен для эксплуатации САПР.

Языки проектирования в свою очередь включают входные, выходные и промежуточные языки.

Входные языки необходимы для задания исходной информации: описания

объектов, описания заданий и процессов. Описание объекта предполагает описание его структуры, свойств, характеристик, взаимодействия между частями объектов, с надсистемой и окружающей средой.

Описание процесса включает в себя описание начальных характеристик процесса, особенностей его выполнения, а также результатов.

Язык описания задания предназначен для идентификации заданий, описания их характеристик и указания последовательности выполнения проектных процедур. Входные языки характеризуются большим разнообразием, однако узкой проблемной ориентацией и изменчивостью при адаптации САПР.

Промежуточные и внутренние языки используются для представления информации на конкретных стадиях проектирования в САПР [44].

Данный набор языков является универсальным, но при этом неудобным при непосредственном использовании проектировщиком. С другой стороны, для встраивания в САПР нового входного языка необходима разработка некоего конвертера данного языка в промежуточный, что предполагает рост временных и трудовых затрат.

В связи с этим актуальной является задача унификации терминологического пространства проектной деятельности посредством объединения входных и промежуточных форм представления терминологии в ЛО САПР с целью упрощения процесса взаимодействия субъектов и объектов проектной деятельности.

1.3 Современные тенденции интеллектуализации САПР

Слабая формализация поставленных задач на начальных этапах проектирования СТС, а также нечеткость описания проблемной области предполагают появление дополнительных трудностей для проектировщиков, решение задач которыми все чаще требует поддержки принятия решений со стороны автоматизированных систем, способных моделировать человеческие рассуждения. В связи с этим, в настоящее время актуальной является задача

разработки интеллектуальных САПР.

Согласно [127, 71], интеллектуальные САПР представляют собой системы автоматизированного проектирования, в которых для повышения эффективности решения проектных задач применяются технологии искусственного интеллекта. В первую очередь это связано с использованием новых методов хранения проектной терминологии, возможностью расширения и обучения декларативной и процедурной составляющей баз знаний, используемых при решении проектных задач, а также с реализацией различных интеллектуальных стратегий поиска проектных решений.

Главными целями интеллектуализации современных САПР являются:

1. Сокращение сроков проектирования.
2. Повышение эффективности и качества проектирования на всех его этапах.

В настоящее время существует две стратегии интеллектуализации САПР: внешняя и внутренняя. Внутренняя стратегия заключается в разработке алгоритмов и методов решения конкретных проектных задач. Внешняя стратегия предполагает разработку интеллектуальных средств управления как всем процессом проектирования, так и отдельными его этапами с возможностью экспертной поддержки в точках принятия решений.

Основными направлениями интеллектуализации САПР являются:

- Использование методов и алгоритмов формализации условий проектных задач, а также трудноформализуемых проектных процедур.
- Использование гибридных моделей хранения проектной терминологии, а также разработка инструментальных средств работы с ней.
- Работа с базами проектной терминологией, обеспечивающих учет опыта решения проектных задач.
- Разработка технологий интеграции САПР с внешними пакетами прикладных программ, посредством унификации моделей представления хранимых знаний.
- Разработка модулей САПР, решающих задачу помощи проектировщику

в принятии проектных решений посредством применения методов анализа и синтеза экспертных знаний [21].

Все эти направления предполагают непосредственное применение методов и алгоритмов искусственного интеллекта.

1.4 Экспертные системы, как элемент интеллектуальной САПР

Проектная деятельность предполагает полное погружение в особенности рассматриваемой предметной области, объем знаний о которой постоянно растет, именно поэтому одной из наиболее значимых и важных задач в области автоматизации процесса проектирования сложных технических систем является создание экспертных систем, как элемента интеллектуальной САПР, позволяющих объединить знания специалистов в данной предметной области и помочь проектировщику в решении поставленных задач, благодаря использованию интеграционных алгоритмов представления знаний.

Согласно [95, 131, 119], «экспертные системы - это сложные программные комплексы, аккумулирующие знания специалистов в конкретных предметных областях и тиражирующие этот эмпирический опыт для консультаций менее квалифицированных пользователей». Обобщенная схема экспертной системы представлена на рисунке 1.2.

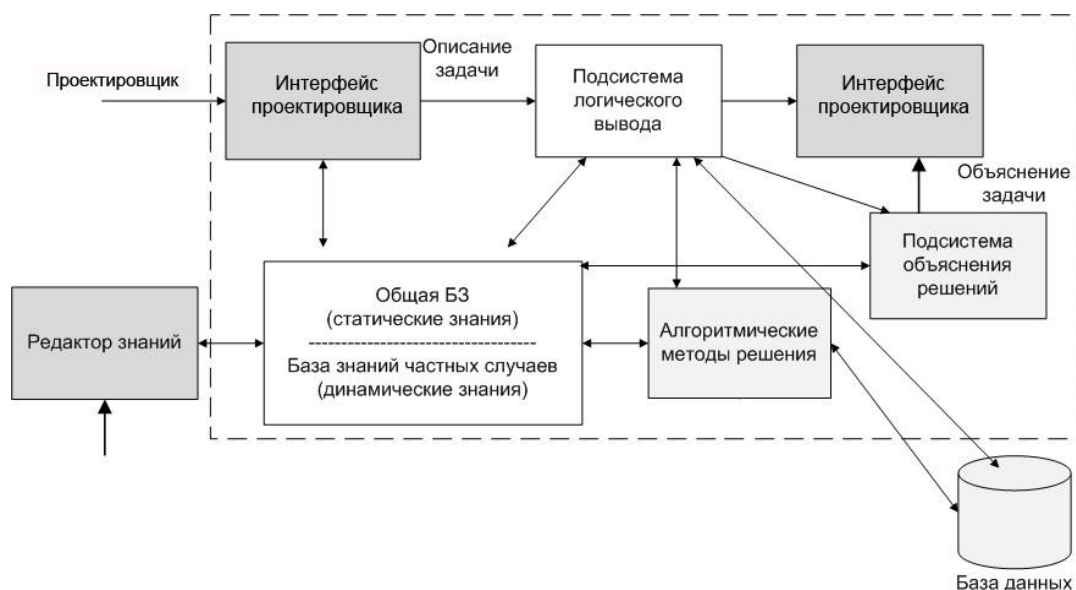


Рисунок 1.2 Структура экспертной системы, как модуля интеллектуальной САПР

Исходя из представленной обобщенной схемы ЭС, как модуля интеллектуальной САПР, основными модулями данной системы являются:

1. Проектировщик — специалист предметной области, для которого предназначена система. Проектировщик нуждается в дополнительной помощи и поддержке своей деятельности со стороны системы.

2. Интерфейс проектировщика — программный модуль, обеспечивающий диалог проектировщика с САПР на стадии ввода информации, а также при получении результатов.

3. Подсистема логического вывода — подсистема, моделирующая ход рассуждений проектировщика на основании хранящихся знаний.

4. Подсистема объяснения решений — программный модуль, позволяющий проектировщику проследить последовательность действий системы в процессе логического вывода. Подсистема отражает набор правил, применимых в рамках решения проектной задачи. Таким образом, исходя из объяснения конкретного решения системой, проектировщик имеет возможность оценить объективность выдачи рекомендации.

5. Алгоритмические методы решения, являющиеся основой работы подсистемы логического вывода.

6. База данных — совокупность вспомогательных, временных данных, участвующих в процессе логического вывода знаний и представленных преимущественно в реляционной форме.

7. Редактор знаний — подсистема, представляющее возможность работы с БЗ в интерактивном режиме.

8. База знаний (БЗ) — ядро системы, совокупность знаний ПрО, представленная в форме, понятной проектировщику (на некотором языке, приближенном к естественному) [51].

Наиболее важными с точки зрения обеспечения основного функционала ЭС являются машина вывода (ризонер) и база знаний. Тенденция интеллектуализации САПР обеспечивает естественный переход от баз данных к базам знаний, которые мыслятся в качестве важнейшего компонента перспективных САПР. Базы знаний

являются логическим продолжением баз данных, они поддерживают и моделируют некоторые элементы интеллектуальной деятельности человека, при этом активно используют механизмы семантической интерпретации [89].

База знаний представляет собой декларативное описание предметной области, и от того, насколько адекватно это описание реальной картине мира, а также насколько универсальна и удобна с точки зрения автоматизированной обработки форма представления предметной терминологии, зависит продуктивность работы экспертной системы. В связи с этим, одной из важнейших задач в области построения ЭС как элементов САПР является поиск необходимых средств, моделей, алгоритмов и форматов представления терминологии в ЛО САПР, удовлетворяющих поставленным перед разработчиком требованиям.

1.4.1 Средства представления терминологии в интеллектуальных САПР

Основными форматами терминологического представления экспертных знаний, используемых в современных интеллектуальных САПР являются:

1. *Онтология, т.е. совокупность «троек»: объект атрибут значение.* Эта форма является переходом от базы данных к базам знаний и позволяет применять приемы, накопленные при работе с СУБД.

2. *Фрейм.* Состоит из именованных ячеек (слотов), в каждую из которых заносится значение. Именно фреймы наиболее часто используются при построении баз знаний в САПР.

3. *Правила продукций: ЕСЛИ <посылка> ТО <заключение>* Традиционная логика включала закон исключения третьего, однако во всех имеющих практический смысл ситуациях нуля и единицы недостаточно для ответа на поставленный вопрос. Поэтому в базах знаний широко используется нечеткая логика, предложенная Лотфи Заде [37, 85].

4. *Семантические сети.* Семантическая сеть - это граф, вершинами которого являются объекты, а дугами логические связи между ними. Семантические связи позволяют представить неограниченно сложные знания, формируя тем самым информационные модели той или иной предметной

области [83, 124].

В настоящее время большинство интеллектуальные САПР используют для хранения и унификации предметной терминологии продукционную модель, позволяющую делать логический вывод рекомендаций с помощью заранее сформированного экспертом набора правил. Недостатком данного метода является неполнота описания предметной области, помимо этого, модель требует строгой структуризации экспертных знаний, что в условиях слабой формализованности данных о предметной области делает использование данной модели оправданным лишь в ограниченных количествах случаев.

Остальные модели представления предметной терминологии позволяют связать результаты анализа с рассматриваемой предметной областью, но не дают возможности делать вывод рекомендаций по результатам анализа вследствие отсутствия встроенного логического вывода в основе моделей.

Использование одного формата представления предметной терминологии существенно снижает гибкость и функциональность экспертных систем поддержки принятия решений, используемых в процессе проектирования сложных технических систем, именно поэтому в настоящее время актуальной является разработка методов и алгоритмов интеграции разных моделей представления слабоформализованных экспертных знаний.

Наиболее универсальной и полной с точки зрения охвата специфики предметной области во всем разнообразии отношений между ее объектами является модель интеграции онтологического и продукционного подходов представления терминологии, позволяющая в процессе логического вывода соответствующих рекомендаций опираться на данные, представленные в виде онтологии и наиболее полно описывающие рассматриваемую предметную область [114].

В работах [22] [40] [8] описаны некоторые предложения объединения онтологий и продукционных правил. Но проблема устранения противоречий в онтологии остается актуальной. Кроме того, в работах используются форматы сохранения правил, не учитывающие особенности проектирования СТС. В

работе [13] предложена концепция использования правил вывода продукционного типа для построения интеллектуальных систем. В вышеуказанных исследованиях не используются все возможности продукционных систем (императивных команд и команд, которые имеют возможность изменять структуру содержания онтологии). В данных работах не предлагается ни методологии, ни фреймворка построения онтологических ЭС (ОЭС) в общем случае, именно поэтому задача интеграции продуктов в онтологическую модель является одной из наиболее актуальных.

1.4.2 Логический вывод в интеллектуальных САПР

На сегодняшний день разработано множество различных методов логического вывода и поиска решений в базе знаний, модулем реализации которых в интеллектуальных САПР является машина вывода (ризонер, решатель) [51, 67]. Определяющим в выборе метода логического вывода является способ представления знаний. Механизм логического вывода (МЛВ) выполняет следующие функции:

- формирование и обработка фактов конкретной ситуации;
- определение порядка выбора и применения фактов и правил. МЛВ

можно представить в виде четырех последовательных процессов:

- выбор активных правил и фактов;
- сопоставление (определяется какие правила выполнять в первую очередь);
- разрешение конфликтов;
- выполнение выбранного означенного правила (действие).

Слабая формализация поставленных задач на начальных этапах проектирования сложных технических систем предполагает необходимость введения элементов теории нечетких систем в описание проблемной области проектных задач, а также применение алгоритмов нечеткого логического вывода с целью поддержки принятия проектных решений.

1.4.3 Нечеткость в логическом выводе знаний

Для решения проблемы формализации качественных характеристик объектов и явлений были введены в процесс логического вывода элементы нечеткости как реализация аппарата нечеткой (fuzzy) алгебры, предложенной американским исследователем Лотфи Заде в начале 70-х годов XX века [133].

В нечеткой логике выделяют три основные группы алгоритмов, которые применяются в современных интеллектуальных САПР в процессе вывода:

1. Алгоритм, работающий по принципу «ситуация-действие». Главной операцией данного алгоритма является поиск методом ближайшего соседа эталонной нечеткой ситуации, которая наиболее соответствует текущей нечеткой ситуации. С этой целью описание текущей ситуации происходит параллельно или последовательно (в зависимости от возможностей) с описанием эталонных ситуаций. В качестве меры сходства может быть использована степень нечеткого равенства или степень нечеткого включения [117, 35].

2. Алгоритм нечеткого композиционного вывода.

За основу данного алгоритма взята операция, максиминной или минимаксной композиции нечеткого множества и нечеткого отношения. Данные композиции представляются в виде матрицы нечетких значений [43, 61].

3. Алгоритм лингвистической аппроксимации.

Целью данного алгоритма является определение составного нечеткого высказывания, частью которого являются значения лингвистических переменных, наиболее соответствующих неизвестному нечеткому множеству, полученному в результате выполнения нечетких операций над исходными данными (например, операций типа композиции). Упрощенно, этот алгоритм можно представить в виде алгоритма поиска (в том числе методом ближайшего соседа) терма, наиболее характерного аппроксимируемому множеству. В качестве меры сходства нечетких множеств может быть использована степень нечеткого равенства, нечеткого включения и другие меры сходства [62, 35].

Согласно [85], «нечетким логическим выводом (НЛВ) является получение заключения в виде нечеткого множества, соответствующего текущим значениям входов, с использованием нечеткой базы знаний и нечетких операций». На

практике в основу НЛВ обычно включают композиционное правило Заде.

В нечетких управляющих системах все правила работают одновременно, но степень их влияния на результат различна. Поэтому основой функционирования нечетких управляющих систем является вычисление обобщенного результата, учитывающего влияние всех правил.

Согласно [37, 85], композиционное правило вывода Заде формулируется следующим образом: «если известно нечеткое отношение \tilde{r} между входной (x) и выходной (y) переменными, то при нечетком значении входной переменной $x = \tilde{a}$, нечеткое значения выходной переменной определяется так:

$$y = \tilde{a} \circ \tilde{r}$$

где \circ - максиминная композиция».

Существует несколько схем логического вывода на базе набора нечетких правил. Рассмотрим некоторые из них.

Предположим, что входная переменная x является нечетким множеством A , и y – нечетким множеством B , тогда выходная переменная z является нечетким множеством C . Таким образом, если задан факт $x = x_0$, $y = y_0$, то необходимо найти $z = z_0$. Схема вывода на базе нечетких правил сводится к решению следующей задачи:

База правил R_1 : если $X - A_1$ и $Y - B_1$, тогда $Z - C_1$;

в противном случае R_2 : если $X - A_2$ и $Y - B_2$, тогда $Z - C_2$;

...

R_n : если $X - A_n$ и $Y - B_n$, тогда $Z - C_n$;

Факт $x = x_0$, $y = y_0$, следствие z . [135, 136]

Отсюда, для проведения логического вывода с использованием базы нечетких правил нужно определить функции:

- представления в виде нечетких понятий (функций принадлежности);
- вычисления логических выражений условных частей правил с логическими связками И, ИЛИ, НЕ;
- вычисления импликации;

- усреднения результата путем композиции.

Основные этапы НЛВ:

1. Фаззификация данных. Фаззификация предполагает интерпретацию четкого значения x_0 в виде нечеткой точки.
2. Композиция входной переменной и условной части правила: $x_0 \circ A_i$, $y_0 \circ B_i$. Если факт задан нечеткой точкой, то на данном этапе производится вычисление соответствующей степени принадлежности.

3. Вычисление нечеткой импликации

$$(x_0 \circ A_i) \wedge (y_0 \circ B_i) \rightarrow C_i \forall R.$$

Результатом выполнения п.2 и 3 для всех правил является n нечетких значений для выхода Z .

4. Агрегация среднего значения:

$$C = \bigcup_{i=1}^n C_i ,$$

Агрегация предполагает построение нечеткого значения выхода по результатам предыдущих этапов.

5. Дефаззификация, т.е. получение четкого значения по агрегированному нечеткому понятию [134].

Общая схема НЛВ представлена на рисунке 1.3

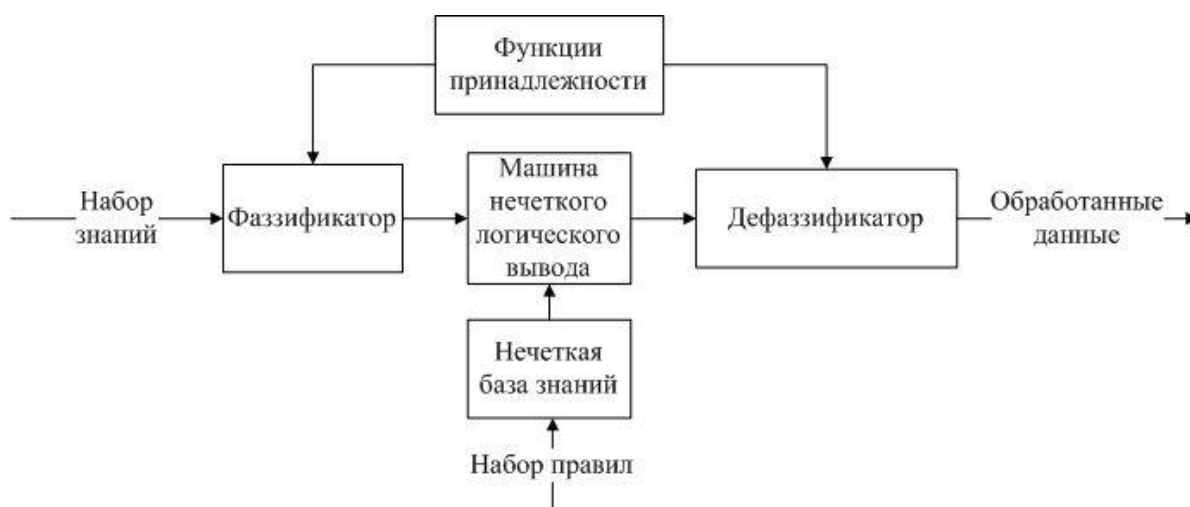


Рисунок 1.3 Общая схема НЛВ

Данный алгоритм НЛВ является базовым. В зависимости от особенностей поставленной перед проектировщиком задачи используется определенная

модификация данного алгоритма. Схемы нечеткого вывода:

1. Алгоритм Мамдани: импликация моделируется минимумом (*min*), а агрегация максимумом (*max*).

2. Алгоритм Цукамото: разработан для монотонных функций. В итоге выходное значение определяется следующим образом:

$$z'_1 = C_1^{-1}(L_1)z'_2 = C_2^{-1}(L_2)z = \frac{(L'_1 z_1 + L'_2 z_2)}{L'_1 + L'_2},$$

где L'_1, L'_2 - уровни достоверности применения правил,

z_1, z_2 – значения выходной переменной по первому и второму правилам.

Ввиду монотонности функций вычисления выходной переменной сводят к усреднению значений, полученных по разным правилам.

3. Алгоритм Суджено: алгоритм ограничивает правые части вывода линейным случаем:

Если $X - A_1$ и $Y - B_1$, тогда $z = a_1x + a_2y$;

Если $X - A_2$ и $Y - B_2$, тогда $z = b_1x + b_2y$.

4. Алгоритм Ларсена: импликация моделируется произведением.

5. Упрощенная схема нечеткого вывода: правые части правил задаются четко:

Если $X - A_1$ и $Y - B_1$, тогда $z = z_i$, где z_i – четкое значение.

В результате вывода получается дискретное множество решений для каждого элемента, для которого задана определенная степень уверенности. В качестве выходной переменной выбирается значение с максимальной уверенностью [54, 88, 91].

1.4.4 Анализ существующих систем, реализующих алгоритм НЛВ

Все рассмотренные выше алгоритмы НЛВ нашли свое применение в совершенно различных областях: в проектировании СТС посредством реализации в качестве модулей в интеллектуальных САПР, так и в управлении СТС, в прогнозировании состояния экономических показателей и технических характеристик и т.д. при расширении функционала отдельных программных

решений [100, 79].

Быстрый рост областей применимости алгоритмов обусловлен главными преимуществами fuzzy-систем:

- возможность работы с входными данными, представленными в нечетком виде;
- возможность использования нечеткого формального описания критериев оценки и сравнения значений характеристик предметной области (например, «большинство», «в основном», «наверное», «скорее всего» и т.д.);
- возможность проведения качественных оценок входных и выходных значений (например, «быстрый рост», «быстрое падение», «высокий» и т.д.);
- возможность работы со степенью достоверности и ее распределением входных данных и выходных результатов;
- возможность моделирования поведения сложных динамических систем и проведение их сравнительного анализа с заданной степенью точности.

В настоящее время множество разработчиков интеллектуальных САПР, а также систем управления сложными техническими объектами, используют в качестве машины вывода модуль, реализующий один из алгоритмов НЛВ.

Примеры интеллектуальных САПР, реализующих различные алгоритмы НЛВ, приведены в таблице 1.1 [87] [42] [58] .

Таблица 1.1 Примеры ИСАПР на основе НЛВ

№	Программная система	Применение	Базовый алгоритм НЛВ
1	Hybrid CAD design environment for mechanical power transmissions (Zarefar, Goulding)	Проектирование трансмиссии автомобилей	Мамдани
2	fuzzy TECH 3.0	Проектирование систем управления	Мамдани, Такаги–Суджено–Канг
3	(M)ANFIS - (multiple) adaptive neuro-fuzzy inference system	Фреймворк используется для проектирования СВЧ-устройств, систем прогнозирования осадков, систем медицинской диагностики и т.д.	Такаги–Суджено
4	САПР экструзионной техники	Проектирование одношнековых экструдеров	Мамдани

	(Зубкова Т.М., Мустюков Н.А., Колобов А.Н.)		
5	CAD-Architecture (Абу Д.М.)	Проектирование радиоэлектронной аппаратуры	Мамдани
6	САПР ИСПДн (Корнев П.А.)	Проектирование систем категорирования персональных данных	Мамдани, Суджено

Помимо реализации НЛВ в качестве модуля ИСАПР, были разработаны программные пакеты, которые могут выступать как в качестве отдельных модулей и фреймворков ИСАПР при проектировании СТС, так и в качестве независимых приложений:

CubiCalc 2.0 RTC (HyperLogic) –система на основе нечеткой логики, на базе которой возможна разработка собственных прикладных экспертных систем. Полученные прикладные решения применяются для управления в условиях неполноты входных данных. С помощью прикладных решений CubiCalc решаются задачи динамического моделирования в предметных областях с параметрами, имеющими качественный характер оценки, а также управления технологическими процессами (ТП).

Rule Maker (HyperLogic) – программная система, предназначенная для первичной обработки входных данных, их кластеризации, поиска взаимосвязей между полученными кластерами и автоматического формирования набора нечетких правил для CubiCalc. При этом границы кластеров также задаются в виде нечетких значений, в связи с этим результат логического вывода на основании полученных правил предполагает наличие степени его достоверности («наверное», «лучше» и т.д.).

FuziCalc (FuzeWare) – представляет собой электронную таблицу, реализующую основные принципы нечеткой логики. Данная программная система служит для проведения вычислений с неточно известными данными без накопления ошибки. Система может быть использована при решении задач анализа и прогнозирования некоторых экономических показателей, состояние которых может быть оценено качественными характеристиками.

Также в таблице 1.2 представлен ряд систем, использующих один или

несколько алгоритмов НЛВ [32, 14, 34]

Таблица 1.2 Примеры экспертных систем на основе НЛВ

№	Программная система	Применение	Базовый алгоритм НЛВ
1	Risk Analysis Expert System	Экспертная система анализа рисков	Мамдани
2	Adaptive-Network-Based Fuzzy	Искусственная нейронная сеть, основанная на нечеткой системе вывода	Такаги-Суджено
3	Fuzzy inference system in river flow forecasting	Система прогнозирования течения реки	Мамдани, Такаги-Суджено-Канг
4	СУ БПЛА	Система управления беспилотными летательными аппаратами	Мамдани
5	FIS To Fault Diagnosis	Система тестирования подшипников в условиях повышения нагрузки	Суджено

Помимо систем, адаптированных к решению ограниченного круга задач, реализация данных алгоритмов может быть проведена в универсальных пакетах и средах математического моделирования Matlab, fuzzyTech, SciLab. Данные программные системы обладают всем инструментарием, необходимым для моделирования конкретных бизнес-процессов, в том числе и в области автоматизированного проектирования, и получения результатов его реализации на основании значений соответствующих характеристик с помощью НЛВ:

MATLAB (+ Fuzzy Logic Toolbox) – данный фреймворк для системы MatLab предназначен для решения задач проектирования нечетких моделей, в том числе и нечетких нейронных сетей, а также для реализации методов нечеткой кластеризации. Помимо этого, фреймворк предоставляет возможность совместной работы с графической средой имитационного моделирования Simulink.

fuzzyTech (INFORM GmbH) – пакет прикладных программ, главным назначением которого является проведение математических вычислений на основе методов нечеткой логики в технической сфере. В отличие от MATLAB, НЛВ в системе fuzzyTech предполагает возможность параллельного или последовательного применения нескольких независимых блоков нечетких

продукций, на основе которых и осуществляется логический вывод.

SciLab – бесплатный пакет прикладных математических программ с открытым кодом, являющийся альтернативой системе MATLAB и решающий аналогичные задачи. Среда SciLab предполагает возможность применения алгоритмов НЛВ по Мамдани и Суджено в процессе решения задач [94].

Несмотря на то, что с каждым годом область применения систем, основанных на использовании алгоритмов НЛВ, расширяется, каждый из рассмотренных алгоритмов имеет значительные недостатки в случае применения к автоматизации проектирования СТС, а именно:

- не учитываются особенности взаимосвязей объектов рассматриваемой предметной области;
- все алгоритмы являются однократными и не предполагают возможности нахождения выхода при изменении входных значений в случае итерационных алгоритмов;
- не учитывается возможная иерархия и синонимия объектов;
- алгоритмы не содержат элементов обучаемости, что значительно снижает потенциал их развития и повышает трудозатраты экспертов на полноценное описание особенностей поведения объектов проектирования.

В связи с этим возникает задача разработки и реализации гибридных алгоритмов логического вывода с использованием нечеткости, обеспечивающих исправление перечисленных недостатков. Одним из таких решений является применение класса нечетких онтологий.

1.5 Нечеткие онтологии

Онтология, будучи формальным отображением реальности, требует четкого представления информации, а реальность не всегда может быть четко выражена. Такое несоответствие порождает сложности концептуального характера. Среди таких сложностей – реализация процедур адекватного НЛВ, реализация адекватной интеграции блоков онтологической информации и прочее. В связи с тем, что объединение различных форматов представления знаний на

онтологической основе является наиболее удачным с точки зрения автоматизированной обработки и восприятия человеком методом формирования базы знаний экспертных систем, одной из наиболее актуальных задач в сфере интеллектуализации САПР является решение проблемы учета нечеткости в рассуждениях проектировщика.

Переход от обычных онтологий к нечетким выглядит вполне естественным, поскольку понятия и отношения естественного языка, представляющие собой исходный материал для построения онтологии неоднозначны, неточны и не имеют жестких границ. Поэтому адекватным средством формализации онтологий могут служить модели на базе лингвистических переменных, нечеткие множества, нечеткие отношения, нечеткие графы и нечеткие деревья, нечеткие ограничения, нечеткие реляционные и алгебраические системы [93].

При интеграции онтологии и систем продукционных правил, проектировщик не получает возможности выбора варианта проектного решения в связи с ограниченностью инструмента логического вывода на основе четких экспертных знаний. В случае же введения в онтологию отдельных наборов нечетких отношений и объектов, в результате вывода проектировщик получает более широкое множество вариантов проектного решения поставленной задачи, каждый из которых обладает определенной мерой уверенности, выраженной соответствующим значением функции принадлежности.

1.5.1 Методика построения нечеткой онтологии из разных источников

Одним из направлений использования нечетких онтологий является формализация особенностей большой предметной области посредством извлечения знаний из различных источников. Чем сложнее формализуемая проблемная область, тем больше внимания необходимо уделять унификации форматов представляемых данных. Помимо этого, при построении онтологии нужно учесть оценку понятий предметной области, исходя из надежности формализуемых источников знаний. Это также возможно реализовать с помощью задания значений соответствующих функций принадлежности, т.е. посредством внесения нечеткости в разрабатываемую онтологию.

На рисунке 1.4 в общем виде представлен фрагмент нечеткой онтологии сложной проблемной области, формируемой из различных источников.

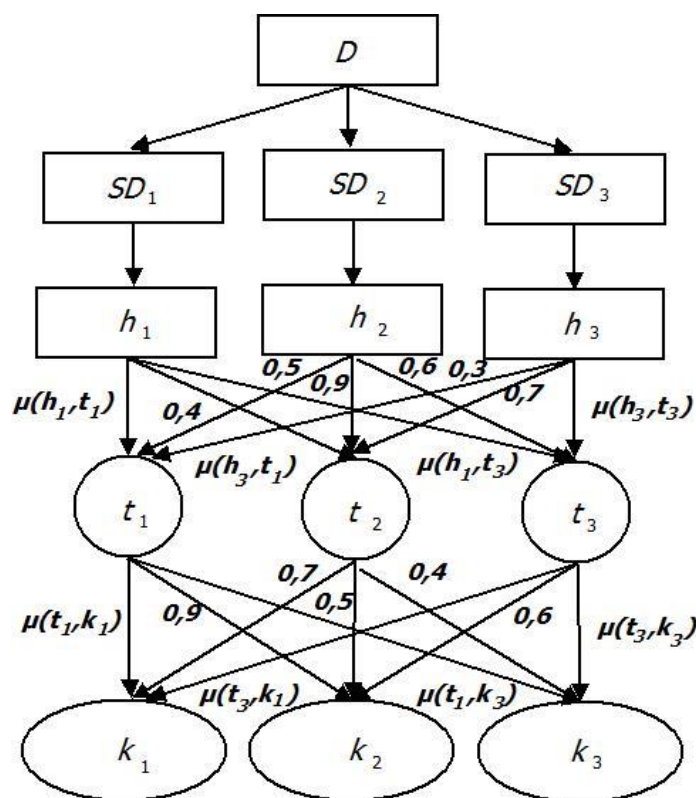


Рисунок 1.4 Фрагмент нечеткой онтологии

Верхний (первый) уровень онтологии предполагает комплексное описание ПрО D , полученное путем слияния знаний из разных источников (экспертов, коллекций текстов). На втором уровне D декомпозируется на подобласти SD , при этом каждой подобласти соответствует определенный источник информации (h_i).

Между базовыми понятиями ПрО (t_i , где $i = [1..n]$) и лингвистическими переменными (k_j , где $j = [1 .. m]$), находящимися на четвертом и пятом уровнях соответственно, устанавливаются семантические отношения, и каждому отношению присваивается значение функции принадлежности $0 \leq \mu(t, k) \leq 1$, где $t \in T_i$, $k \in K_j$.

Помимо этого, в процессе построения нечеткой онтологии ПрО производится соотнесение категорий t_i ПрО с источниками h_i , из которых данные категории могли быть извлечены. Подобным образом формируется набор ключевых слов для каждого источника знаний, а также для каждой подобласти комплексной ПрО, и устанавливаются значения степеней принадлежности между ними, выражаемые

числами из интервала $[0, 1]$ ($\mu(h, t)$).

Таким образом, множество объектов, категорий и понятий онтологии, лингвистических переменных, а также значений функций принадлежности образуют нечеткие множества категорий формируемой онтологии [126].

1.5.2 Расширение OWL с помощью FuzzyOWL

В настоящее время существует наиболее универсальный и имеющий достаточную поддержку со стороны разработчиков интеллектуальных экспертных систем подход к построению нечетких онтологий - методология FuzzyOWL. Синтаксис FuzzyOWL предполагает три формата используемого алфавита: для представления нечетких классов (fuzzy concepts), нечетких отношений (fuzzy roles) и нечетких объектов классов онтологии (fuzzy individuals). В отличие от OWL DL, реализующей разновидность дескрипционной логики SHOIN(D), концепция FuzzyOWL охватывает все положения логики SROIQ(D):

- S – это набор базовых классов, имен свойств и имен объектов, является подмножеством DL в связке с транзитивными свойствами;
- R – возможность определения составных аксиом вложенности ролей в наборе аксиом перечисленных видов : $R \circ S \subseteq R$, $R \circ S \subseteq S$, где $R \circ S$ – композиция ролей;
- O – наличие номиналов (так называемых классов единичных объектов);
- I – наличие обратных свойств;
- Q – качественные ограничения кардинальности ролей;
- (D) – типы данных, свойства, которые связывают объекты с типами данных [6].

1.5.3 Свойства-аннотации OWL

Основополагающим элементом методологии построения нечетких онтологий с точки зрения структуры OWL, является наличие свойства аннотации fuzzyLabel, которое хранит параметры и значения функции принадлежности, соотносимые с конкретным классом, объектом, отношением внутри заданного нечеткого множества данных [5].

Свойства-аннотации OWL могут использоваться для добавления информации для классов, отдельных индивидов и свойств объектов/типов данных. OWL допускает аннотации с различной информацией и метаданными классов, свойств, индивидов и онтологий.

Разновидность языка OWL – OWL DL, используемая в экспертных системах для дальнейшего получения знаний из уже существующих в онтологии, накладывает ряд ограничений на использование свойства аннотации — двумя из наиболее важных ограничений являются:

1. наполнителем для свойства аннотации должны быть символьные данные, ссылки URI или индивидуумы;
2. в свойствах аксиом нельзя использовать свойства аннотации, например, они не могут использоваться в иерархии свойств. Также для них не должны устанавливаться домен и диапазон [18].

1.5.4 Формальная модель нечеткой онтологии FuzzyOWL

В общем виде любая FuzzyOWL-онтология может быть представлена следующим образом:

$$I = (I_f, C_f, P_f, A_f, D_f, Q_f, L_f, Mod_f),$$

где

- I_f – множество объектов классов онтологии;
- C_f – множество нечетких классов онтологии:

$$C_f = \{C_f^A, C_f^C\},$$

где C_f^A - множество абстрактных классов, C_f^C - множество конкретных классов онтологии;

- P_f — множество свойств объектов:

$$P_f = \{P_f^A, P_f^C\},$$

где P_f^A множество конкретных свойств, т.е. свойств объектов (ObjectProperty),

P_f^C - множество абстрактных свойств, т.е. свойств типа данных (DatatypeProperty);

- D_f — множество аксиом онтологии:

$$D_f = \{A_f^{ABox}, A_f^{TBox}, A_f^{RBox}\},$$

где A_f^{ABox} — множество утверждений об индивидах, A_f^{TBox} — множество терминологических аксиом, A_f^{RBox} — множество аксиом отношений (иерархия отношений). Часть аксиом может быть подклассом множества нечетких аксиом, которые предполагают истинность утверждения с определенной степенью.

- O_f — множество значений степеней, которые могут быть добавлены к объекту нечеткой аксиомы:

$$O_f = \{LD_f, MD_f, ND_f, Var_f\},$$

где LD_f — множество лингвистических переменных, MD_f — множество степеней модификации, ND_f — множество численных значений степеней, Var_f — множество переменных.

- L_f — множество операторов нечетких логик соответствующих типов.

$$L_f = \{L_f^{Luk}, L_f^{Zad}, L_f^{Goed}, L_f^{Prod}\},$$

где L_f^{Luk} — множество операторов логики Лукасевича, L_f^{Zad} — множество операторов логики Заде, L_f^{Goed} — множество операторов логики Геделя, L_f^{Prod} — множество операторов продукционной логики. Определяются с помощью встроенного отношения *hasSemantics*;

- Mod_f — множество «нечетких модификаторов», т.е. функций модификации функций принадлежности, нечетких классов и нечетких отношений. Функции могут быть линейными или треугольными. В таблице 1.3 приведены элементы нечетких аксиом FuzzyOWL, а также их возможное представление.

Таблица 1.3 Элементы нечетких аксиом в FuzzyOWL

№	Элемент	Возможные значения	Представление в FuzzyOWL
---	---------	--------------------	--------------------------

1	LD_f – множество лингвистических значений степеней	«рост», «спад», «высокий», «низкий»	$\langle \text{AnnotationAssertion} \rangle$ $\langle \text{AnnotationProperty}$ $\text{IRI} = \text{"\#fuzzyLabel"} / \rangle$ $\langle \text{IRI} \rangle \text{"\#ВысокаяНагрузка"} \langle / \text{IRI} \rangle$ $\langle \text{Literal}$ $\text{datatypeIRI} = \text{"\&rdf;PlainLiteral"} \rangle \text{fuzzyOwl2}$ $\text{fuzzyType} = \text{"datatype"};$ $\text{Datatype type} = \text{"rightshoulder"}; a = \text{"15.0"};$ $b = \text{"30.0"}; / \rangle \text{fuzzyOwl2} \langle / \text{Literal} \rangle$ $\langle / \text{AnnotationAssertion} \rangle$
2	MD_f – множество степеней модификации	«очень», «быстрый», «медленный»	$\text{type} = \text{"modified"} \text{ modifier} = \text{"очень"}$
3	ND_f – множество численных значений степеней	$0 \leq ND \leq 1$	Degree Value=0,6
4	Var_f – множество переменных	a, b, c, k1, k2	$b = \text{"30.0"};$
5	L_f – множество операторов нечетких логик соответствующих типов	Zadeh, Lukasiewicz Goedel and Product	$\text{hasSemantics} = \text{"Zadeh"}$
6	$Modf$ – множество нечетких модификаторов	Linear, Triangular	$\langle \text{Datatype type} = \text{"triangular"} \text{ a} = \text{"32.0"}$ $b = \text{"41.0"} \text{ c} = \text{"50.0"} / \rangle$

Алгоритм задания нечеткого отношения, согласно методике FuzzyOWL, выглядит следующим образом:

1. Создается новое свойство-аннотация fuzzyLabel, в котором будут задаваться параметры нечеткости каждого элемента онтологии;
2. Лингвистические переменные определяются в форме новых созданных типов данных (Datatype).
3. Для каждого созданного типа данных задаются верхнее и нижнее границы принимаемых значений, выбирается тип функции принадлежности и для нее задаются соответствующие параметры. Схематично данная последовательность действий представлена на рисунке 1.5.

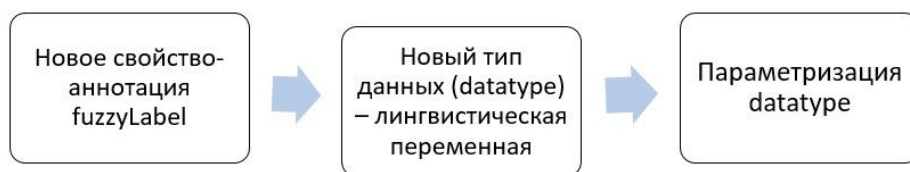


Рисунок 1.5 Схема задания нечеткого отношения в нотации FuzzyOWL

Методология FuzzyOWL предполагает аналитическое определение нескольких возможных типов функций принадлежности:

- Треугольные (triangular);
- Трапецевидные (trapezoidal);
- Линейные (linear);
- Кусочно-линейные: лево и правосторонние (left-shoulder and rightshoulder) [29].

Таким образом, нечеткие алгоритмы могут применяться в рамках интеллектуальных САПР, в частности для выдачи рекомендаций по принятию проектных решений в трудноформализуемых ситуациях в конкретной предметной области.

1.6 Методики извлечения терминов в задачах расширения ядра онтологии

Одним из направлений использования онтологических структур в САПР является индексирование технической документации на основе онтологий, содержащих ограниченный набор отношений между категориями – тезаурусов. Каждому объекту (категории) тезауруса приписывается некоторый пояснительный текст, таким образом регламентируя между ними отношения подчиненности, синонимии, иерархии и т.д. Структура подобных индексных тезаурусов регламентируется следующими стандартами:

- ГОСТ 7.25-80 [75] - для одноязычных тезаурусов;
- ГОСТ 7.24-2007 [74] - для многоязычных тезаурусов;
- ISO 2788.

Главной особенностью онтологий (в том числе и индексных тезаурусов) является необходимость её развития и обновления, вследствие изменения представляемой ПрО. Расширение онтологии - сбор знаний, опирающийся на (полу-) автоматические методы для преобразования неструктурированных, слабоструктурированных и структурированных данных в концептуальные структуры.

На данный момент существует несколько направлений расширения ядра онтологии посредством:

- извлечения знаний из интернет-ресурсов (в частности, wiki-ресурсов) [84, 71];
- анализа словарей и тезаурусов [72, 130];
- мерджинга (слияния) онтологических структур [68, 101];
- извлечения терминологии в процессе обработки текстов по тематике рассматриваемой предметной области с использованием статистических, лингвистических и семантических методов [101, 72, 50].

Извлечение терминологии («term extraction») представляет собой анализ текста на естественном языке с последующим формированием перечня терминов-кандидатов с целью расширения словарной базы, тезауруса, онтологии и т.д. Эта операция позволяет избавиться от терминологической избыточности и добиться последовательности терминологии, а потому имеет большое значение в управлении терминологией (terminology management) в задачах автоматизированного пополнения базы знаний интеллектуальных САПР с целью наиболее развернутого и подробного описания рассматриваемой предметной области [98, 97].

Принцип работы существующих алгоритмов извлечения терминологии основан на статистических и лингвистических методах. Статистические методы предполагают оценку степени терминологичности на основе статистики встречаемости соответствующего словоупотребления.

В основе лингвистических методов лежит отбор по определенным лексико-грамматическим шаблонам и другим лингвистическим признакам термина [2].

Главным недостатком использования статистических и лингвистических методов в процессе извлечения терминологии из текста с целью расширения используемой в процессе автоматизированного проектирования базы знаний является отсутствие возможности выделения из получившегося множества терминов только тех, которые относятся к рассматриваемой проблемной области. Именно для решения подобных задач используют семантические алгоритмы,

базирующиеся на определенных семантических метриках [3].

Рассмотрим основные алгоритмы извлечения терминологии из больших объемов текстов с целью расширения ядра предметной онтологии.

Лингвистический и статистический методы

В работах, посвященных лингвистическому подходу к извлечению терминологии [7, 72, 99], утверждается, что для извлечения терминологии достаточно морфо-синтаксической информации.

Существует две формы реализации лингвистического подхода к извлечению терминологии из текстов:

- Морфологический анализ текста, включающий:
 - определение частей речи слов в словосочетаниях и соответствующих им морфологических признаков;
 - приведение слов к начальной форме (лемматизация);
 - определение лексико-грамматических классов в тексте.
- Извлечение терминологии в соответствии с заданными лексико-грамматическими классами.

В целях повышения качества извлечения терминологии в алгоритм могут добавляться дополнительные методы. Например, для отсекаемых определенных словоупотреблений, которые не могут являться ни термином, ни частью термина, можно использовать стоп-лист, состоящий из таких словоупотреблений. Чаще всего стоп-лист составляется экспертом на основе частотного словаря, построенного автоматически для релевантной предметной области [52].

Предварительно полученная терминологическая информация (в т.ч. глоссарии терминов, словари, контрастные корпуса, тезаурусы в САПР и др.) может использоваться в качестве образца и быть полезной как в процессе извлечения терминологии, так и в процессе фильтрации [33].

Разработанный лингвистический метод включает отбор слов и словосочетаний согласно лексико-грамматическим шаблонам, выделенным

экспертом в результате анализа предметной области. Рассматриваются следующие шаблоны: N, N+N, N+N+N, A+N, A+A+N (N – существительное, А – прилагательное) и другие. Отбор предваряется морфологическим анализом текста, в результате которого выясняется информация о каждом словоупотреблении в тексте.

Суть использованного лингвистического метода заключается в рассмотрении всех анализируемых в данном тексте слов и словосочетаний и принятии решения об отнесении их к выделенным экспертом лексико-грамматическим классам. Каждый лексико-грамматический класс рассматривается в результате обхода с исключением. Каждые М слов (М – количество слов в лексико-грамматическом классе) проверяются по маске лексико-грамматического класса. Если комбинация слов удовлетворяет маске, то каждое из них помечается как использованное, что обеспечивает вхождение слова только в один лексико-грамматический класс, в то время как словосочетание относится к соответствующему лексико-грамматическому классу. Классы рассматриваются, начиная с классов с большим количеством словоупотреблений, заканчивая классами с меньшим их количеством [53].

Рассмотрим основные статистические методы извлечения терминологии.

Frequency

$$f_{xy}$$

где f_{xy} – частота биграммы xy .

Применение метода Frequency [9] предполагает вычисление абсолютных частот всех двусловий в анализируемом тексте. Предполагается, что наиболее часто встречаемые в тексте двусловия семантически характеризуют текст, поэтому их можно отнести к классу терминов для данной предметной области.

В силу своей простоты данный метод входит в число наиболее значимых методов извлечения терминологии [9, 17], при этом является базовой частью для более сложных статистических методов, основанных на встречаемости слова в рассматриваемом тексте или корпусе. Среди недостатков метода выделяется отсутствие учёта связи между словами, таким образом, метод может срабатывать

неверно при анализе двусловий, относящихся к общеупотребительной лексике и, как следствие, не имеющих отношения к предметной области (тематике) рассматриваемого текста или корпуса.

T-Score

$$\frac{f_{xy} - \frac{f_x f_y}{n}}{f_{xy}^2}$$

где f_x, f_y – частота x, y соответственно; n – количество биграмм в корпусе.

Метод T-score [63] основан на методе Frequency и предназначен для исправления его недостатка, связанного с отсутствием учёта связей между словами в двусловии. Таким образом, метод представляет собой численную оценку взаимосвязи двух слов. Эту оценку следует понимать, как выражение степени доверия, с которой можно утверждать, что именно эти два слова, употреблённые вместе, представляют собой термин. Однако недостатком данного метода является то, что он основан на несвойственных естественному языку нормально распределённых вероятностях.

T-score приближает биномиальное распределение дискретной случайной величины к распределению, близкому к нормальному распределению непрерывной случайной величины. При этом за нулевую гипотезу принимается гипотеза о независимости. Следовательно, данный метод имеет недостатки, связанные с допущением о нормальном распределении [11].

T-score представляет собой модифицированное ранжирование двусловий по частоте как самого двусловия, так и составляющих это двусловие одиночных слов. Очевидна прямая зависимость величины от частоты двусловия в коллекции. К сожалению, преимущества данной величины сводятся только к коррекции результатов для наиболее частотных двусловий [132].

Mutual Information (коэффициент взаимной информации)

$$MI = \log_2 \frac{f(x, y) \times N}{f(x) \times f(y)}$$

Мера MI [4] вычисляет коэффициент встречаемости двух слов. Разница между величинами произведения относительных частот каждого слова с величиной их совместной встречаемости выявит степень значимости их встречаемости.

Если $MI(x; y) > 1$, тогда сочетание слов статистически значимо, если $MI(x; y) \approx 0$, то слова появляются в паре крайне редко, если же $MI(x; y) < 0$, то x и y находятся в отношении дополнительной дистрибуции.

Для данной меры используется порог отсечения по частоте [86]. При этом значение порога зависит от задачи и объема корпуса: высокий порог обусловлен задачей выделения наиболее значимых, характерных для данной коллекции словосочетаний.

Очевидно, что мера принимает тем большее значение, чем чаще слова встречаются вместе; однако мера может неверно сработать с низкочастотными словосочетаниями. Для более полного анализа биграмм следует устранить эту проблему.

Log-Likelihood

С целью устранения недостатка метода Mutual Information был использован оптимизированный метод Log-Likelihood, заключающийся в предположении о биномиальном характере функции распределения совместной встречаемости слов. Рассчитывается по формуле [80]:

$$\begin{aligned} \loglikelihood = & a * \log(a+1) + b * \log(b+1) + c * \log(c+1) + d * \log(d+1) - \\ & (a+b) * \log(a+b+1) - (a+c) * \log(a+c+1) - (b+d) * \log(b+d+1) - (c+d) * \log(c+d+1) \\ & + (a+b+c+d) * \log(a+b+c+d+1), \end{aligned}$$

где a - частота данного словосочетания (пары); b – суммарная частота других (отличных от данной) пар с той же самой левой леммой; c – суммарная частота других пар с той же самой правой леммой; d – суммарная частота пар, отличных от данной и не попадающих в категории (b) и (c).

C-value и NC-value

$$C-Value(a) = \begin{cases} \log_2 |a| * freq(a), & \text{если не вложен} \\ \log_2 |a| * (freq(a) - \frac{1}{P(T_a)} * \sum_{b \in T_a} freq(b)) & \end{cases}$$

a – кандидат в термины,

$|a|$ – количество слов в словосочетании,

$freq(a)$ – частота встречаемости a ,

T_a – множество словоупотреблений, в которых содержится кандидат в термины a ,

$P(T_a)$ – количество словоупотреблений, в которых содержится кандидат в термины a .

Метод выделения многословных терминов, предложенный Frantzi et al., выделяет значимые словосочетания, которые не являются составной частью более длинных словосочетаний.

C-value – метод извлечения терминов, являющихся частью более длинного терминологического словоупотребления [12]. Метод C-value использует данные лингвистического анализа (части речи, лингвистические шаблоны для отделения определенного типа терминов, стоп-лист) наряду со статистическим анализом.

С одной стороны, метод C-value обогащает статистические подходы к извлечению терминологии через направленность на определенный тип терминов – вложенные термины. С другой стороны, метод NC-value создает новые преимущества для C-value: 1) метод извлечения, учитывающий слова, являющиеся частью контекста термина (слова, которые тяготеют к совместной встречаемости с терминами), 2) использование информации о словах, являющихся частью контекста термина, для извлечения многословных терминов [39].

Так как метод C-value включает в себя лингвистическую составляющую, выбор языкового фильтра влияет на точность и полноту списка вывода, различные языковые фильтры будут иметь влияние на точность выдачи. NC-value является расширением C-value, который включает в себя информацию в контексте извлечения терминологии, что позволяет более полно судить о степени терминологичности двусловия. Мера NC-value рассчитывается по следующей формуле [12]:

$$NC - Value = 0,8C - Value(a) + 0,2 \sum_{b \in C_a} f_a(b)weight(b)$$

где a – кандидат в термины,

C_a – множество двусловий, отличных от a ,

$f_a(b)$ – частота двусловия b , входящего в множество C_a ,

$weight(b)$ – вес двусловия.

Примечательно, что NC-value также зависит от лингвистической составляющей, так как в его расчёт входит вычисление C-value.

Частотность с учетом частоты употребления в объемлющей коллекции (TF*IDF)

Данная статистическая мера [24][38] часто применяется в информационно-поисковых системах (ИПС) и позволяет снижать вес употребительных слов. В ИПС данная мера используется для оценки релевантности документа поисковому запросу; может применяться в задачах анализа документов и кластеризации как мера близости документов.

$$Tf*Idf(w) = Tf*log((n-b)/b)$$

где n – размер контрастной коллекции, b – число документов, в которых употреблялось слово w в контрастной коллекции.

Статистическая мера TF*IDF может использоваться для оценки важности слова в данном документе, который, в свою очередь, является частью корпуса текстов или, по крайней мере, обширной коллекции документов. Данная оценка пропорциональна частоте слова в рассматриваемом документе, в то время как обратно пропорциональна частоте слова в корпусе или коллекции.

TF (Term Frequency – частота слова) – численное значение, равное отношению количества вхождений какого-либо слова в документ к числу слов в нём. Иначе этот множитель можно интерпретировать как важность слова в пределах рассматриваемого документа. TF может быть представлена следующей формулой:

$$TF = \frac{w_i}{\sum_{k=1}^n w_k},$$

где w_i – частота i -го слова в документе; n – количество слов в документе.

IDF (Inverse Document Frequency — обратная частота документа) – инверсия частоты встречаемости слова в документах коллекции или корпуса [25] [38]. Использование данного показателя позволяет отличать широкоупотребительные слова в качестве нетерминов или терминов другой предметной области, нерелевантных данной, рассматриваемой в коллекции или корпусе. IDF соответствует следующая формула:

$$IDF = \log\left(\frac{n - b}{b}\right),$$

где n – размер контрастной коллекции, b – число документов, в которых употреблялось слово в контрастной коллекции.

Z-score

Значение меры Z-score [19] представляет собой стандартное отклонение между ожидаемой частотой встречаемости u рядом с x и наблюдаемой частотой:

$$Z - score = \frac{O_{11} - E_{11}}{\sqrt{E_{11}}}$$

где E_{11} – ожидаемая частота биграммы, O_{11} – наблюдаемая частота биграммы;

Величина Z-score тем больше, чем больше совместная частота встречаемости двух словоупотреблений. В отличии T-score, мера Z-score чаще всего применяется в тестах на больших выборках [23].

Chi-squared (χ^2)

χ^2 -критерий Пирсона [64] предназначен для анализа таблиц сопряженности 2x2. Четыре значения, формирующие таблицу, – это:

- 1) частота данного двусловия;
- 2) частота двусловий с участием первого слова (но не второго);
- 3) частота двусловий с участием второго слова (но не первого);
- 4) Частота всех остальных двусловий.

В качестве меры расхождения берется значение:

$$\chi^2 = \sum_{i=1}^4 \frac{(n_i^* - n_i)^2}{n_i}$$

где n_i^* – наблюдаемая частота; n_i – ожидаемая частота (в соответствии с предположением о случайности сочетания слов).

В качестве ожидаемых значений берутся маргинальные частоты. Двусловия упорядочиваются по убыванию значения χ^2 .

Отношение функций правдоподобия (LR)

В данном методе используется отношение функций правдоподобия, соответствующих двум гипотезам – о случайной и неслучайной природе двусловия [65]. Логарифм отношения функций правдоподобия выглядит следующим образом:

$$\log \lambda = \log \frac{b(c_{12}, c_1, p)b(c_2 - c_{12}, N - c_1, p)}{b(c_{12}, c_1, p_1)b(c_2 - c_{12}, N - c_1, p_2)}$$

где $b(k, n, x)$ – значение биномиального распределения для k успешных исходов в n независимых испытаниях при вероятности успешного исхода в каждом испытании, равном x ; c_1 – частота первого слова двусловия; c_2 – частота второго слова двусловия; c_{12} – частота двусловия; N – длина текста; $p = c_2/N$; $p_1 = c_{12}/c_1$; $p_2 = (c_2 - c_{12})/(N - c_1)$.

Двусловия упорядочиваются по возрастанию значения $\log \lambda$.

1.7 Постановка задачи исследования

При анализе существующих методов представления терминологии в лингвистическом обеспечении САПР был выявлен ряд проблем:

- необходимость унификации процесса автоматизированной обработки терминологии в лингвистическом обеспечении при решении проектных задач различного рода;
- необходимость разработки семантического базиса анализа представления терминологии в рамках решения задач автоматизированного проектирования;
- отсутствие интегративных концептуальных моделей, использующих различные подходы представления знаний о предметной области в форме тезауруса;

- необходимость одновременного использования разноаспектных описаний особенностей рассматриваемой предметной области;
- необходимость решения проблемы учета опыта решения схожих проектных задач;
- необходимость адаптации и расширения лингвистического и информационного обеспечения проектной деятельности вследствие изменения особенностей ПрО;
- необходимость решения проблемы учета нечеткости в рассуждениях проектировщика.

Отсюда главной целью данного исследования является разработка и реализация эффективных моделей и алгоритмов представления терминологии в лингвистическом обеспечении САПР на основе интеграции нечетких онтологий и логического вывода, обеспечивающих снижение количества ошибок проектировщика.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести сравнительный анализ современных средств представления терминологии в лингвистическом обеспечении интеллектуальных САПР;
- разработать онтологически-ориентированную систему извлечения терминологии из проектных документов с использованием сформированного ядра предметной онтологии;
- провести сравнительный анализ современных интеллектуальных методов логического вывода в САПР, выявить их возможности и ограничения в плане решения задач автоматизированного проектирования;
- рассмотреть возможность применения методов онтологического анализа для решения задач оценки состояния сложного технического объекта рассматриваемой предметной области с целью поддержки принятия проектных решений на экспертном уровне;
- исследовать возможность интеграции методов теории нечетких систем и онтологического анализа состояния сложной технической системы;
- разработать онтологическую модель представления набора продукций

и алгоритм логического вывода рекомендаций на экспертном уровне в задачах автоматизированного проектирования;

- разработать алгоритм расширения процедурной составляющей базы знаний интеллектуальной САПР с использованием механизма прецедентов;

- разработать алгоритм расширения декларативного описания ПрО с использованием сформированного ядра онтологии интеллектуальной САПР посредством извлечения терминологии из проектных документов;

- разработать автоматизированную систему (АС) поддержки проектирования, реализующую алгоритм интеграции механизмов логического вывода и онтологического представления экспертных знаний с элементами нечеткости, а также использующей механизм учета прецедентов с целью расширения ядра онтологии;

- провести вычислительные эксперименты, позволяющие оценить эффективность предложенных моделей и алгоритмов в процессе проведения концептуального проектирования сложных технических систем.

При решении задачи оценки эффективности предложенных моделей и алгоритмов необходима адаптация условий проведения экспериментов под специфику решаемых задач.

Глава 2 Методы и алгоритмы интеграции онтологического анализа и логического вывода

2.1 Выбор класса модели онтологии для САПР

Главной особенностью предметных онтологий является необходимость её перманентного развития и обновления вследствие постоянного развития как конкретного проекта, так и представляемой предметной области. Основной методикой развития онтологии как одной из форм представления предметной терминологии является интеллектуальное извлечение семантики из больших объемов неструктурированных данных, чаще всего – из текстов, проектных документов.

Извлечение семантики из текстов включает в себя технологии, направленные на извлечение смысловых единиц, фактов, атрибутов, концепций и событий для того, чтобы на выходе проектировщик получил структурированный набор данных, с которым можно проводить дальнейший анализ предметной области при проектировании.

Следовательно, анализ предметной области, предполагающий вывод новых знаний на основании уже имеющихся, включает в себя следующие этапы:

1. Формирование ядра предметной онтологии.
2. Расширение ядра онтологии с использованием семантических алгоритмов извлечения терминологии из текстов предметной области.
3. Логический вывод новых знаний посредством интеграции онтологии и системы правил.
4. Переход к п.2 с использованием в качестве ядра уже имеющейся онтологии (Рис. 2.1).

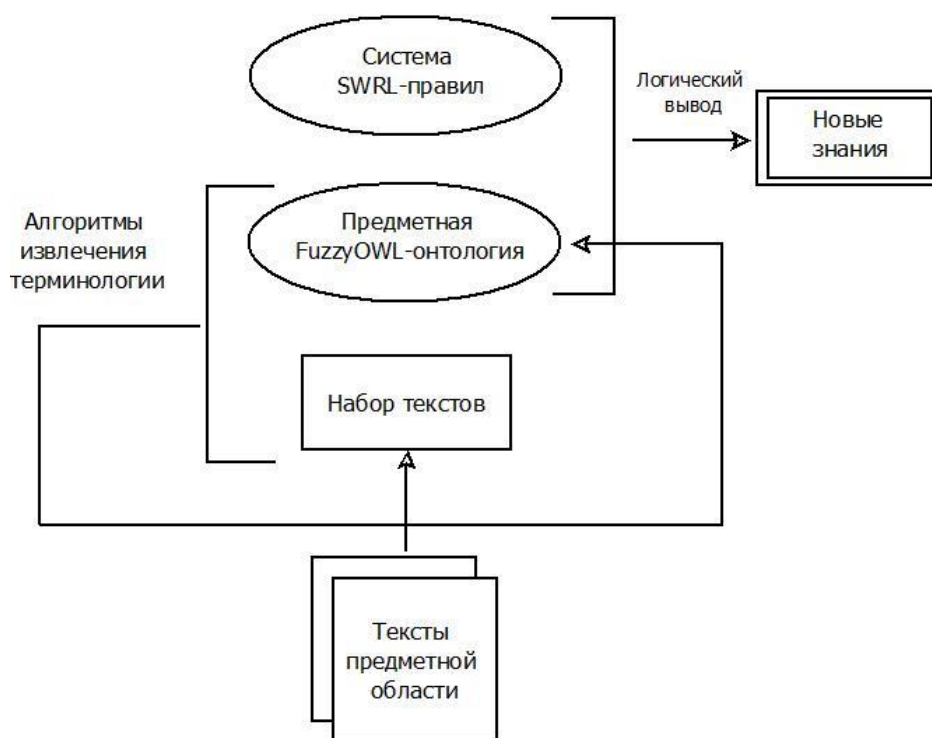


Рисунок 2.1 Схема гибридизации самообучения онтологии и логического вывода

Онтология, согласно [99]- это «формализация некоторой области знаний с помощью концептуальной схемы». Онтология, в классическом понимании, состоит из классов, объектов данных классов, отношений между классами и объектами, а также аксиом, задающих логические ограничения на все элементы онтологии.

Глоссарий представляет собой словарь терминов некоторой предметной области с определениями и, возможно, примерами и переводом. Таким образом, глоссарий тоже является подклассом онтологических структур, однако, с пустым множеством отношений между объектами.

Онтология с единственно определенным отношением «род-вид» между элементами является таксономией. В отличии от таксономий, в *рубрикаторах* смысл отношений между разными рубриками (классами, объектами) может различаться [73].

В *информационно-поисковых тезаурусах*, помимо таксономических, может быть определен набор дополнительных отношений (синонимия, антонимия, и др.), необходимых для решения задач информационного поиска [120]. Структура наполнения подобных тезаурусов регламентируется стандартом ГОСТ 7.24-2007

[74]. Подобные онтологические ресурсы также называют «легкими онтологиями» («lightweight ontologies»).

По мере повышения глубины формализации предметной области, усложняется и структура онтологической модели, на основе которой и строится ее представление.

Обязательной составляющей сложных онтологических структур является наличие логических утверждений, называемых аксиомами, которые позволяют определять элементы онтологии путем логического вывода. Для описания подобных сложных онтологических структур применяются различные логики (дескрипционные логики, логика предикатов первого порядка, модальные логики и др.). Наиболее популярными языками представления онтологий, реализующих механизмы подобных логик, являются:

- RDF «Resource Description Framework» — это разработанная консорциумом Всемирной паутины модель для представления данных, в особенности — метаданных. RDF представляет утверждения о ресурсах в виде, пригодном для машинной обработки.

- OWL — Web Ontology Language, стандарт W3C, язык для семантических утверждений, разработанный как расширение RDF и RDFS;

- KIF (англ. Knowledge Interchange Format — формат обмена знаниями) — основанный на S-выражениях синтаксис для логики;

- Common Logic (CL) (англ.) — преемник KIF (стандартизован - ISO/IEC 24707:2007);

- СусL (англ.) — онтологический язык, использующийся в проекте Сус. Основан на исчислении предикатов с некоторыми расширениями более высокого порядка;

- DAML+OIL (англ.) - является языком преемником DAML и OIL, который сочетает в себе черты обоих. В свою очередь, был заменен на Web Ontology Language (OWL) [71]. Стандартизирован W3C в марте 2001 г.

2.2 Формальная модель онтологии ПрО

Структуру онтологии ПрО, обеспечивающей интеграцию с продукционной формой хранения знаний, запишем в виде кортежа [139]:

$$O = (T, R^D, F),$$

где

T – термины прикладной области, которую описывает онтология. Множество терминов представлено в виде:

$$T = \{C, In_C\},$$

где C – множество классов онтологии, In_C – множество объектов классов онтологии.

R^D - множество отношений между объектами онтологии:

$$R^D = \{R_{In}, R_F, R_{InF}, R_{Tran}, R_{Sym}, R_{Asym}, R_{Ref}, R_{Irref}\},$$

где

- R_{In} – множество обратных отношений: если некоторое свойство связывает объект a с некоторым объектом b , то его обратное свойство связывает объект b с объектом a ;
- R_F – множество функциональных отношений: если свойство является функциональным, то для данного объекта может существовать не более одного объекта, который имеет отношение к первому индивиду через это свойство;
- R_{InF} – множество обратных функциональных отношений: если свойство является обратным функциональному свойству, то это значит, что для конкретного объекта может быть множество объектов, относящиеся к первому объекту через это свойство;
- R_{Tran} – множество транзитивных отношений: если отношение транзитивное и отношение связывает объекта a и объекта b , а также объекта b связывает с объектом c , то объект связан с объектом c через это отношение;
- R_{Sym} – множество симметричных отношений: если отношение p симметричное, и отношение связывает объект a с объектом b , то объект b связан также с объектом a через отношение p ;
- R_{Asym} – множество асимметричных отношений: если отношение p асимметричное, и отношение связывает объект a с объектом b , то объект b не

может быть связан с объектом a через отношение r ;

- R_{Ref} – множество рефлексивных отношений: отношение r называется рефлексивным, если объект связан сам с собой;
- R_{Irref} – множество иррефлексивных отношений: если отношение r иррефлексивное, то оно связывает объект a с объектом b , где объект a объект b обязательно разные;

F – множество аксиом онтологии, задающих логические ограничения на классы, объекты и отношения онтологии.

Множество интерпретирующих функций (аксиом) представлено в виде:

$$F = \{F_C, F_{In}, F_R\},$$

где

$F_C: \{C_{Rule}\} \rightarrow \{C\}$ – функция, сопоставляющая набору классов, используемых в системе продукционных SWRL- правил онтологии множество классов OWL- онтологии;

$F_{In}: \{In_{Rule}\} \rightarrow \{In\}$ – функция, сопоставляющая набору объектов классов, используемых в системе продукционных SWRL- правил онтологии множество классов объектов OWL-онтологии;

$F_C: \{R_{Rule}\} \rightarrow \{R^D\}$ – функция, сопоставляющая набору отношений объектов и типа данных, используемых в системе продукционных SWRL- правил онтологии множество отношений классов OWL-онтологии [108].

2.3 Алгоритм интеграции онтологии и логического вывода

В рамках данной диссертационной работы были исследованы особенности продукционного и онтологического подходов к представлению знаний соответствующей предметной области, а также был разработан алгоритм интеграции онтологической модели с помощью использования набора SWRL- правил [30].

В общем виде любое правило можно представить следующим образом:

$$a_1 \wedge a_2 \wedge \dots \wedge a_n \rightarrow b,$$

где $a_1...a_n$ – предикаты (атомы) антецедента (условия); b – консеквент

(следствие) правила, состоящий из одного атома.

В целом, SWRL – это технология, которая помогает описать абстрактный механизм оперирования объектами предметной области, а также закономерности предметной области. Основным достоинством SWRL является то, что он дает возможность выводить новые факты из существующих утверждений [70].

Продукционные правила SWRL представляют собой простейшие логические структуры, связывающие условия (антецедент) со следствием (консеквентом). Представление рассматриваемой области знаний в виде причинно-следственных связей объясняется тем, что рекомендации и выводы в области анализа состояния сложной технической системы имеют форму составных условных высказываний. Схема простейшего SWRL-правила представлена на рисунке 2.2.

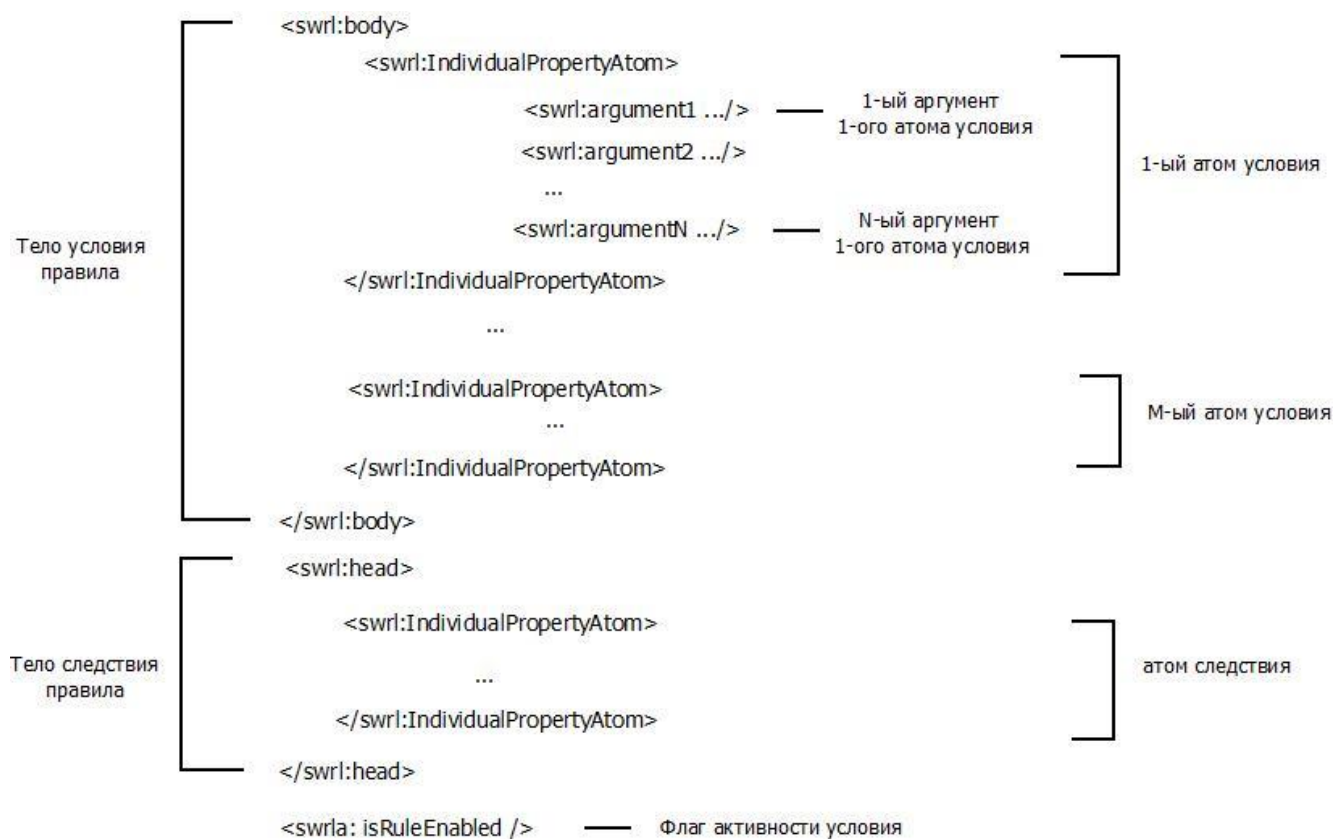


Рисунок 2.2 Общая схема SWRL-правила

SWRL-правила представляют собой дизъюнкты Хорна и в общем виде представлены выражением:

$$Ax : C_1(?x) \wedge C_2(?y) \wedge r_1.(?x, ?y) \wedge C_3(?x, ?z) \rightarrow C_2(?z, ?y),$$

где $(C_1, C_2, C_3) \in C$, $r_1 \in R$, x, y – экземпляры или переменные, z – переменные или значения.

Для примера возьмем предметную область анализа состояния локальной вычислительной сети в процессе искусственного повышения трафика. Тогда пример представления одного из SWRL-правил будет выглядеть таким образом:

*Ситуация(?x) \wedge включаетПоказатель(?x, Фоновая_нагрузка) \wedge
имеетТенденцию(Фоновая_нагрузка, Рост) \wedge
включаетПоказатель(?x, Утилизация_канала_связи) \wedge
имеетТенденцию(Утилизация_канала_связи, Спад) \wedge
включаетПоказатель(?x, Число_коллизий_в_сети) \wedge
имеетТенденцию(Число_коллизий_в_сети, Рост) \rightarrow предполагает(?x,
Проблема_архитектуры)*

Модель представления экспертных знаний в виде набора SWRL-правил имеет ряд преимуществ перед остальными подобными технологиями:

1. правила SWRL не содержат конкретных объектов, а только ссылаются на них, что дает возможность применять одно и то же правило к ряду групп объектов;
2. правила SWRL могут быть добавлены к OWL-описанию, т. е. включены в онтологию;
3. написание и «чтение» правил удобнее, если для этого существует специальный язык.

Схема взаимодействия owl-онтологии и набора SWRL-правил представлена на рисунке 2.3 [102].

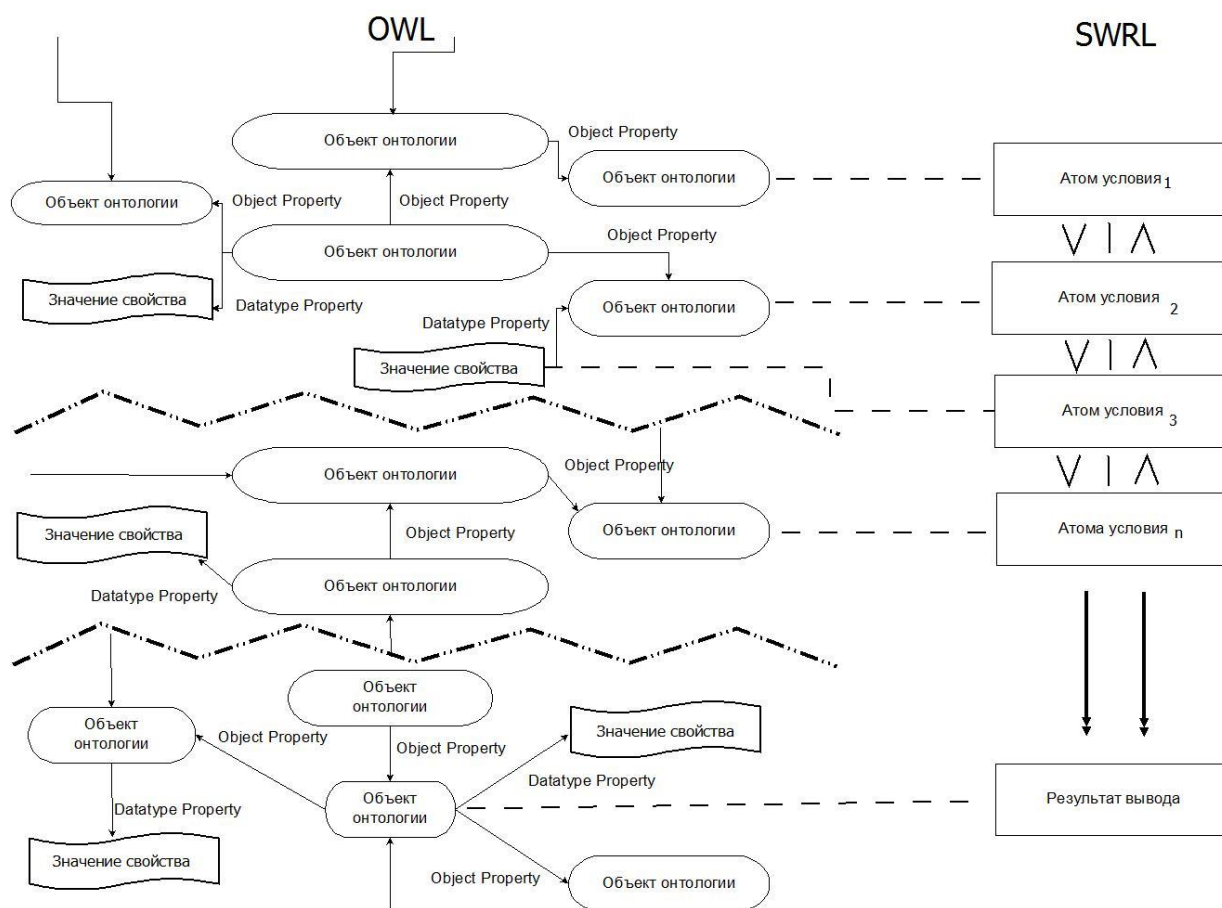


Рисунок 2.3 Схема взаимодействия owl-онтологии и набора SWRL-правил

На примере рассмотренной выше предметной области, набор данных, составляющих атомы SWRL-правил, представляется следующим образом:

$$I = (C, D, P, G, S, H),$$

где

C – множество технических характеристик рассматриваемой системы («Топология_сети», «Маршрутизация» и т. д.);

D – множество характеристик работоспособности системы («Время_реакции_ПО», «Загрузка_процессора_сервера» и т. д.);

P – множество свойств объектов («включаетПоказатель», «имеетТопологию» и т. д.);

G – множество свойств типа данных («имеетПропускнуюСпособность», «имеетЗначение» и т. д.);

S – нечеткое множество характера тенденции («Спад», «Рост» и т. д.);

H – множество вспомогательных терминов, обеспечивающих взаимодействие объектов, используемыми в SWRL – правилах, с индивидами онтологии

(«Ситуация» и т. д.).

Набор правил, которые обеспечивают логический вывод результатов анализа, можно разделить на две группы:

1. Группа продукционных правил, консеквентом (следствием) которых являются конкретные рекомендации по корректировке структуры ЛВС, являющиеся объектом класса «Рекомендации» онтологии предметной области. Модель подобного правила представлена в виде выражения :

$$p_1(x, d) \wedge p_2(d, s) \wedge \dots p_i(x, c_1) \wedge g(c_1, c_2) \wedge \dots \rightarrow p_n(x, h)$$

2. Группа правил, следствием которых является логическое присвоение одному из объектов-характеристик сети конкретного значения. Далее это значение выступает в роли входных данных других продукционных правил в процессе генерации логического вывода рекомендации. Модель подобного SWRL-правила представлена в виде выражения :

$$p_1(x, d) \wedge p_2(d, s) \wedge \dots p_i(x, c_1) \wedge g(c_1, c_2) \wedge \dots \rightarrow p_n(x, c_n) \wedge g(c_n, c_{n+1}),$$

где $x, h \in H$, $p_1, p_2, \dots, p_i, \dots, p_n \in P$, $g, \dots, g_n \in G$, $d \in D$, $c_1, c_2, \dots, c_n, c_{n+1} \in C$.

Схематично описанные виды правил представлены на рисунке 2.4.

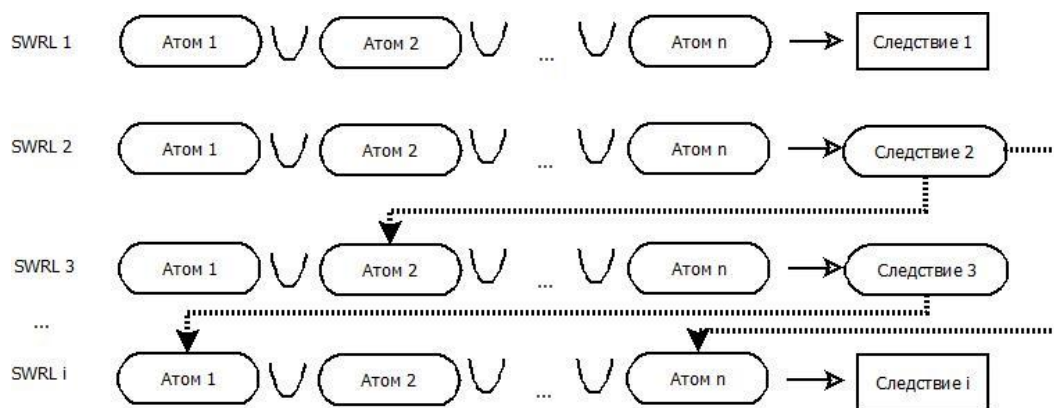


Рисунок 2.4 Группы SWRL-правил

Непосредственный процесс интеграции онтологической и продукционной моделей знаний предполагает использование в качестве объектов SWRL аналогичные объекты классов, описанных в построенной OWL-онтологии.

Взаимосвязь самой онтологии предметной области и набора SWRL-правил в процессе логического вывода осуществляется с помощью формирования запросов к онтологии, генерируемых системой анализа при выполнении набора правил.

Базовый алгоритм формирования вывода на основе интеграции онтологии и систем продукции представлен на рисунке 2.5.

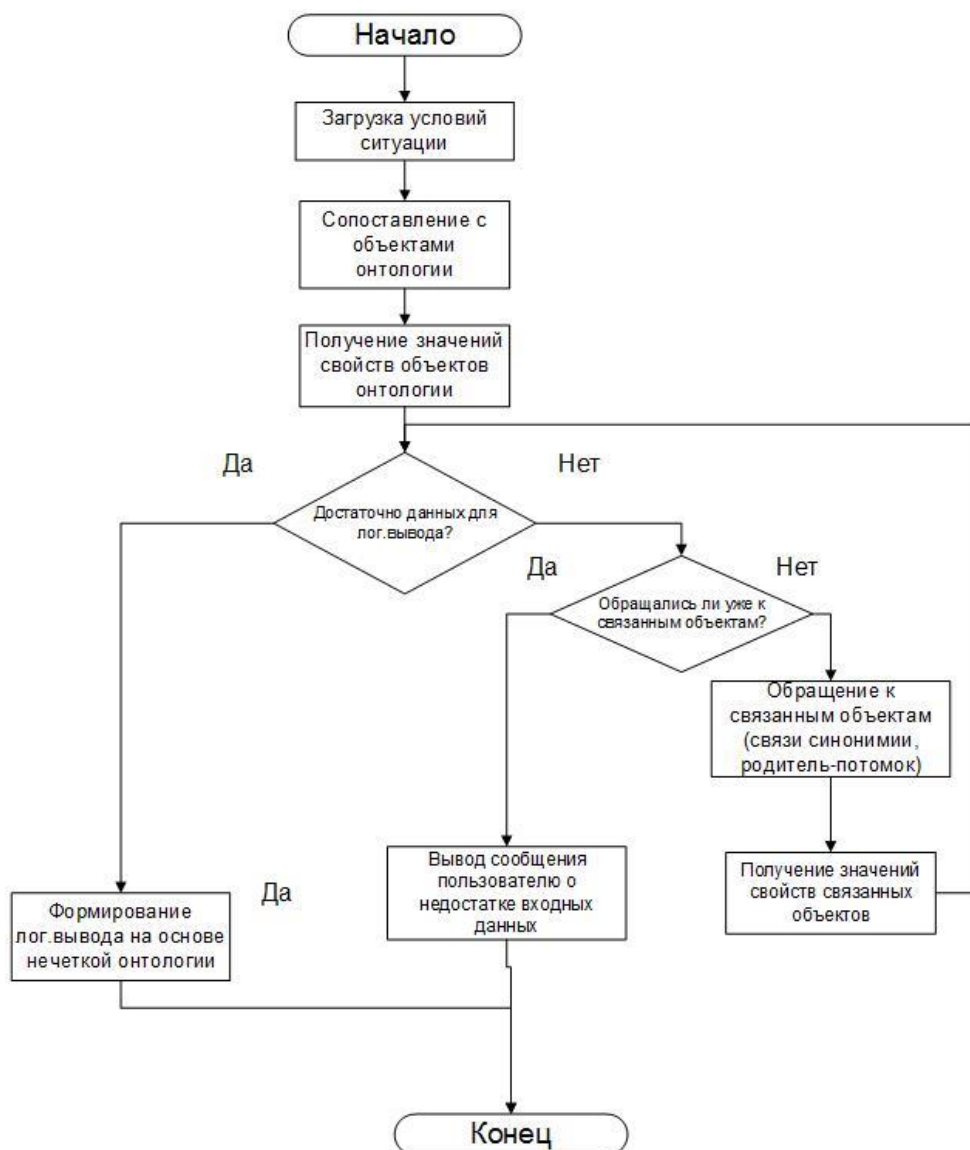


Рисунок 2.5 Базовый алгоритм логического вывода на основе онтологии

Основная последовательность действий в процессе логического вывода рекомендации на основе взаимодействия OWL-онтологии и системы продукционных SWRL-правил включает в себя следующие этапы:

1. Проверка входных данных на наличие в онтологии соответствующих классов, отношений, объектов;
2. Определение лингвистических переменных и соответствующих значений функции принадлежности по численному значению входного параметра, определенного в онтологии;
3. Запрос на выполнение набора правил анализа ситуации и вывода

рекомендации;

4. Запрос на получение связанных данных (обращение к родителям объектов);
5. Запрос на выполнение правил с учетом полученных связанных данных;
6. Вывод результата анализа (одна или несколько рекомендаций) в виде объекта разработанной OWL-онтологии с учетом показателя степени уверенности S .

Логический вывод рекомендаций с использованием онтологии, содержащей нечеткие отношения, классы, а также объекты классов, имеет существенные отличия по сравнению с извлечением знаний из четкой онтологии (см. рисунок 2.6) [112].

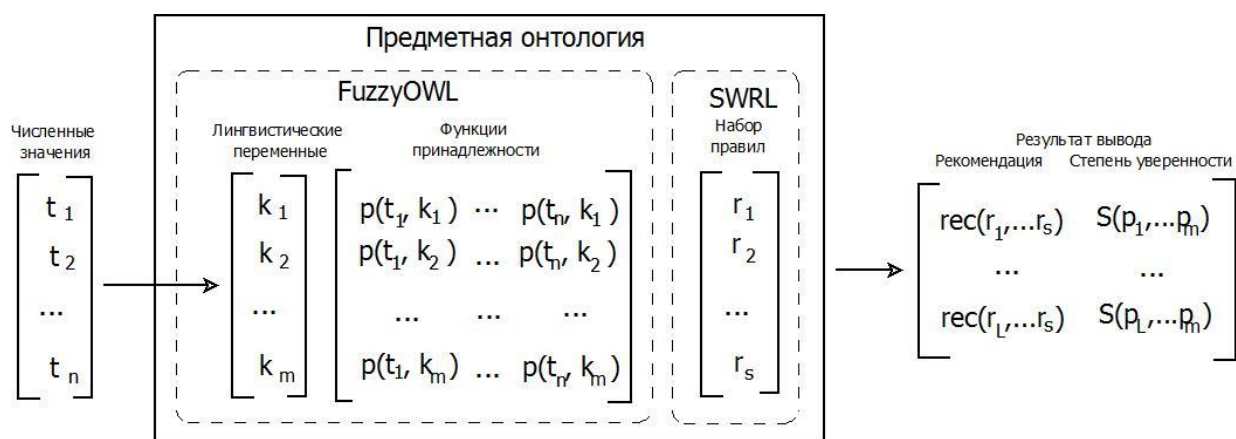


Рисунок 2.6 Схема логического вывода на основе нечеткой онтологии

Основная последовательность действий в процессе логического вывода рекомендации на основе взаимодействия нечеткой FuzzyOWL-онтологии и системы продукционных SWRL-правил включает в себя следующие этапы:

1. На вход алгоритма поступают значения характеристик анализируемой сложной системы. Данные характеристики могут носить качественный или количественный характер, а также непосредственно задаваться в виде нечеткой характеристики значения («большой», «малый», «быстрый рост» и т.д.). В этом случае, в процессе логического вывода в онтологии проводится непосредственный поиск аналогичных объектов классов без процесса лингвистической оценки значения.
2. Композиция входных значений и условных частей SWRL-правил.

В правила, помимо нечетких отношений и объектов, могут включаться и конкретные численные значения соответствующих показателей, а также наименования, четкие отношения и другие характеристики, заданные в онтологии.

Значениям лингвистических переменных, заданным в теле SWRL - правила, соответствуют некоторые степени принадлежности, определенные в онтологии, и в дальнейшем формирующие некую величину степени уверенности в процессе вывода активного правила.

3. Импликация множества пар – «рекомендация – показатель степени уверенности».

При этом показатель степени уверенности является функцией от поступающих значений функций принадлежности, относящихся к заложенным в соответствующем правиле значениям предметной области:

$$S_{Rule} = \min \mu(t_i; k_j),$$

где t – термин проблемной области, присутствующий в сработавшем SWRL- правиле; k – лингвистическая переменная; $j=[1..m]$; $\mu(t,k)$ – значение соответствующей функции принадлежности [111].

Таким образом, решение по исправлению той или иной ситуации, моделируемое значениями входных параметров и получившее наивысшую оценку, будет иметь значение степени уверенности равное 1.

В конечном же итоге, проектировщик получит на выходе набор рекомендаций на естественном языке, каждой из которых будет поставлено в соответствие некоторое значение $S = [0..1]$.

Таким образом, процесс вывода представляет собой движение резонера по графу, в узлах которого находятся объекты соответствующих классов онтологий, удовлетворяющих или не удовлетворяющих заданным условиям.

2.4 Использование прецедентов в процессе логического вывода

При решении сложных задач в области автоматизированного проектирования проектировщик должен получать всестороннюю поддержку со стороны системы, имея возможность выбора решения не только посредством использования экспертных знаний, хранящейся в базе системы, но и с учетом опыта решения данной задачи специалистами в прошлом. Такой опыт может быть формализован в виде набора прецедентов, которые должны обрабатываться и учитываться в рамках решения задачи параллельно основному алгоритму.

Согласно [10], «рассуждения по прецедентам (case-based reasoning, CBR) — это метод формирования умозаключений, опирающийся не на логический вывод от исходных посылок (логические рассуждения), а на поиск и анализ случаев формирования подобных умозаключений в прошлом».

С точки зрения процесса проектирования, механизм прецедентов предполагает поиск подобных задач в базе знаний, хранящей прошлый опыт проектирования, и логический вывод решений на их основе. Применение механизма прецедентов при решении проектных задач оправдано только при выполнении набора условий, относящихся к особенностям проблемной области.

Постановка задачи проектирования предполагает необходимость наличия подобных решений в базе проектных знаний (принцип регулярности). Так как проектная деятельность является строго регламентированным процессом, то задачи, с которыми сталкивается проектировщик, имеют тенденцию к повторению, что предполагает высокую вероятность того, что решение текущей либо будущей проектной задачи имеет решение в прошлом опыте. [26].

Процесс рассуждений по прецедентам может быть схематически представлен в виде цикла. В работе [1] приведено описание типового процесса рассуждений по прецедентам, который состоит из этапов:

- получение прецедента новой задачи (прецедента запроса);
- поиск похожих прецедентов в проектной базе знаний в соответствии с условиями проектной задачи;
- повторное использование решений найденных прецедентов и их адаптация к новой задаче;

- проверка предлагаемого решения на адекватность;
- сохранение нового решения в БЗ.

Логический вывод проектных заключений в интеллектуальных компонентах САПР, использующих механизм прецедентов и основанных на применении набора правил, предполагает выбор в результате сопоставления на основе знаний, хранящихся в фоновых структурах, наиболее универсальными из которых являются онтологии.

Принципиальные различия между методиками логического вывода на основе систем продукций и прецедентов были в полной мере изложены в [16]. Базовые отличия данных методик, важные с точки зрения автоматизации проектирования:

1. Правила представляют собой шаблоны, т.е. содержат переменные и не описывают непосредственно решение, прецеденты же в свою очередь являются константами и оперируют конкретными объектами базы знаний.

2. Применение механизма прецедентов предполагает адаптацию готового решения проектной задачи, взятого из БЗ. Логический вывод с применением набора продукций представляет собой цикл, на каждой итерации которого могут быть получены отдельные значения характеристик системы, не являющиеся готовым решением задачи.

3. Правило из БЗ выбирается путем точного сопоставления условия и входных данных, в отличие от прецедентов, выбор которых проводится путем частичного сопоставления [78].

2.5 Формальная модель системы вывода, основанной на анализе прецедентов

Модель модифицированного алгоритма логического вывода с учетом механизма прецедентов, может быть представлена следующим образом:

$$CBR = \{Cases, I, S_{CASE}\},$$

где Cases – база прецедентов;

I – онтология предметной области (нечеткая);

S_{CASE} – алгоритм поиска подходящего прецедента.

Структура прецедента может быть представлена в следующем виде:

$$Case = \{Index_{CASE}, D(Index_{CASE}), Eff(D(Index_{CASE}))\},$$

где

$Index_{CASE}$ – индекс прецедента, т.е. описание начальной ситуации;

$D(Index_{CASE})$ – множество решений поставленной задачи;

$Eff(D(Index_{CASE}))$ – множество оценок эффективности принятого решения задачи.

Так как прецеденты, в отличие от правил, оперируют не переменными, а конкретными объектами классов, а также значениями свойств этих объектов, то, согласно (1), начальная ситуация описывается следующим множеством:

$$Index_{CASE} = \{ I_f, P_f \},$$

где I_f – множество объектов классов онтологии;

P_f – множество значений свойств соответствующих объектов [110].

Аналогична и модель решения задачи: решением является объект вспомогательного класса «Рекомендации» разработанной онтологии, который в качестве вывода в процессе работы алгоритма выдает лингвистическое значение свойства типа данных (DatatypeProperty) «имеетОписание» выбранного объекта класса «Рекомендации».

В ходе исследования разработана модификация алгоритма логического вывода с использованием прецедентов для задач автоматизированного проектирования.

Для внесения в базу знаний начального набора прецедентов может использоваться универсальный редактор OWL-онтологии Protégé 4.x. Данная программная система предполагает возможность представления знаний в форме:

- нечетких онтологий (с использованием Fuzzy OWL Plugin);
- SWRL-правил;
- набора прецедентов в OWL-формате.

Для обработки прецедентов в OWL-формате в настоящее время используется java-фреймворк jColibri. Данный набор библиотек является бесплатным и свободно распространяемым. Корректный анализ прецедентов посредством использования фреймворка jColibri предполагает необходимость создания в OWL-онтологии трех базовых классов:

CBR-CASE - класс, содержащий экземпляры прецедентов;

CBR-DESCRIPTION - класс, описывающий ограничения, накладываемые на прецеденты;

CBR-INDEX - хранит структуру прецедента, т.е. классы-потомки и их экземпляры, участвующие в процессе поиска подходящего прецедента.

Главное особенностью модифицированного алгоритма вывода рекомендаций (рис. 2.7) является проведение параллельного и независимого логического вывода результатов анализа на основании базы правил и базы прецедентов [113].

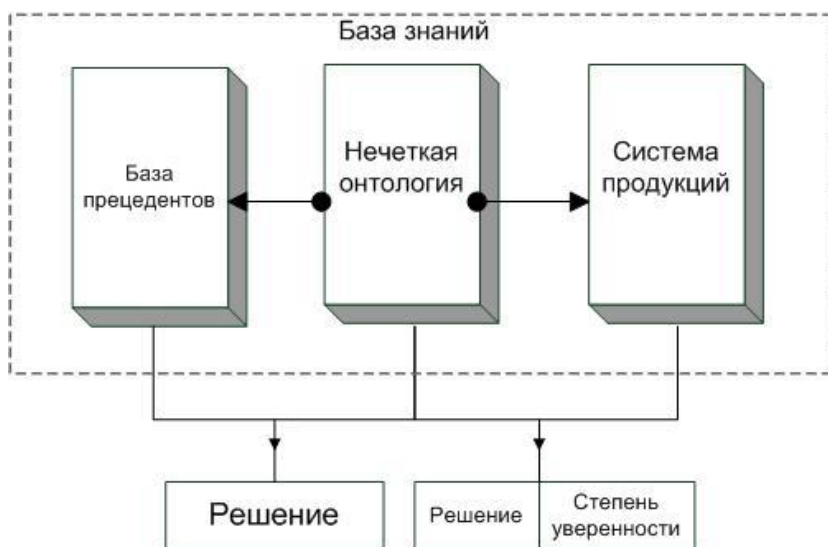


Рисунок 2.7 Схема модифицированного алгоритма логического вывода с использованием базы прецедентов

Таким образом, пользователь получает возможность принятия решений как с учетом анализа закономерностей конкретной области, так и на основании опыта пользователей, уже сталкивавшихся с подобными задачами.

Помимо этого, наличие базы прецедентов позволяет внести элемент обучаемости данного алгоритма.

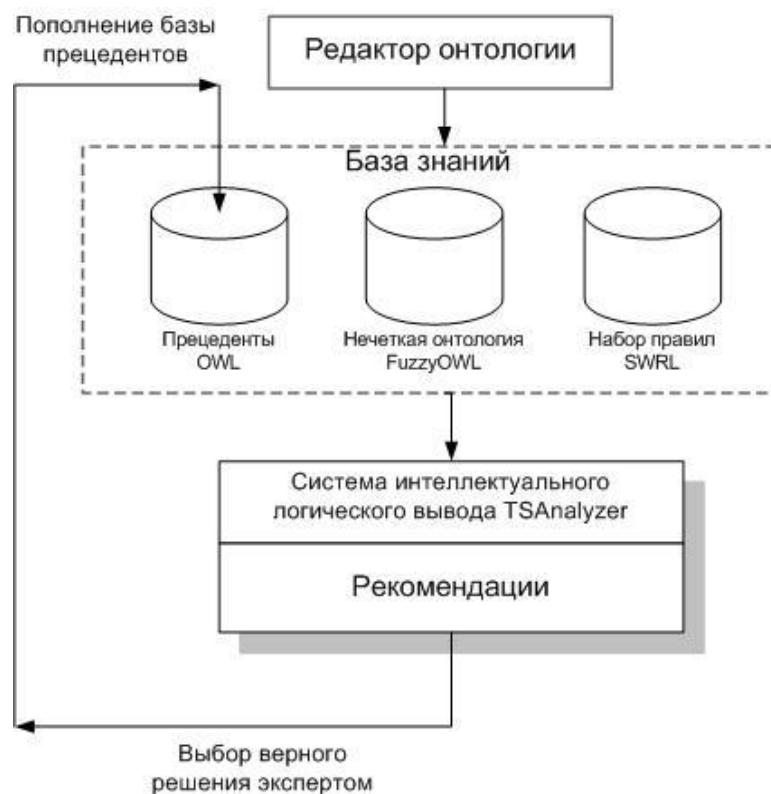


Рисунок 2.8 Схема алгоритма формирования базы прецедентов

Главным преимуществом алгоритма формирования базы прецедентов, схема которого представлена на рисунке 2.8, является возможность проектировщика по выбору решения, который оказался верным при решении подобной задачи [36].

Проектное решение, полученное в результате логического вывода на основе базы продукционных правил и выбранное проектировщиком в качестве правильного, будет занесено в базу прецедентов с исходными условиями задачи. В результате данного действия, при решении аналогичной задачи в следующий раз проектировщик получит эту рекомендацию в качестве априорной, т.е. полученной на основе анализа опыта решения подобной задачи.

2.6 Алгоритмы расширения онтологии

Рассматриваемая в данной работе задача извлечения терминологии из текста по предметной области связана с задачей пополнения и поддержки лингвистического обеспечения САПР. Лингвистическое обеспечение САПР представляется совокупностью языков, используемых при автоматизированном проектировании, терминов и определений, правил формализации естественного языка, а также методов сжатия и развертывания текстов (ГОСТ 22487-77) [82].

В задаче проектирования процесса производства продукции часто возникает потребность в документации, релевантной процессу производства готовой продукции и эксплуатации средств производства. Такая документация составляется, как правило, на основе некоторого глоссария, тезауруса и т.д. предметной области. Любая предметная область, в свою очередь, может быть иметь характерную терминологию, понимание которой доступно узкому кругу специалистов [118].

В то же время в процессе проектирования всего жизненного цикла производства сложного продукта возникает потребность в унификации используемой терминологии с целью обеспечения связности процесса производства. Эта задача усложняется участием в процессе специалистов разных компетенций и уровней подготовки, а также различными форматами используемой документации.

Таким образом, терминология предметной области в составе лингвистического обеспечения САПР имеет большое значение в процессе проектирования, т.к. она обеспечивает проект словарной базой и является основной для проектной и сопутствующей документации. Здесь возникает задача извлечения терминологии из текста по предметной области. Извлечение терминологии из текстов по предметной области имеет целью формирование аппарата терминов и понятий, формирующих лингвистическое обеспечение САПР, в виде словарей, тезаурусов и т.д.

Получение адекватного результата вывода рекомендации экспертной системы интеллектуальной САПР напрямую зависит от правильности и полноты представления предметной области в разработанной онтологии. И чем шире область, тем больше времени и сил экспертов требуется на ее разработку. В связи с этим в настоящее время одной из наиболее актуальных задач в рамках решения проблемы интеграции различных форматов представления знаний является максимальная автоматизация процесса построения онтологии конкретной предметной области.

2.6.1 Тезаурусный алгоритм

Тезаурусный алгоритм извлечения терминологии на множестве слов конкретного текста с использованием ядра OWL-онтологии вычисляет значения степень семантической близости входных словосочетаний к терминам ПрО. Предлагаемый алгоритм позволяет выделить из массива поступающих однословий/многословий только те термины и сочетания, которые относятся к данной предметной области.

Степень семантической близости входного сочетания слов к терминам ПрО k_{ont} может иметь значение от 0 до 1: чем ближе полученное значение к 1, тем с большей долей уверенности данное одно-/многословие является термином.

Тезаурусный алгоритм предполагает непосредственный поиск вхождений лемм поступающих на вход слов и их сочетаний среди терминов, определенных в онтологии. Для этого в онтологии для каждого класса необходимо определить свойство «имеетЛемму», которое имеет строковое значение, полученное путем лемматизации (приведения к начальной форме) имени объекта с помощью программы Mystem компании Яндекс по соответствующим морфологическим признакам термина.

Тезаурусный алгоритм состоит из следующих этапов:

1. оценка степени близости поступающего на вход алгоритма слова/сочетания слов каждому объекту онтологии;
2. поиск опорного объекта ядра OWL-онтологии, наиболее близко ассоциирующегося с входным одно-/многословием [56].

Схема разработанного алгоритма представлена на рисунке 2.9.

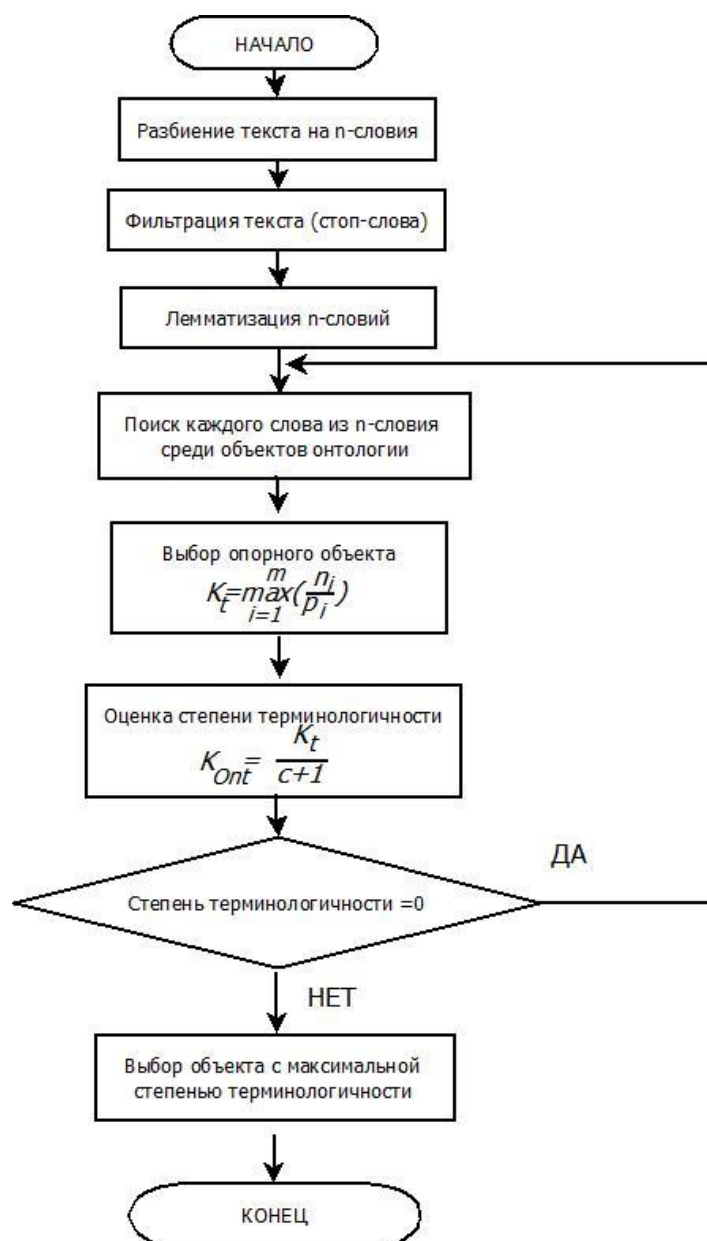


Рисунок 2.9 Блок-схема тезаурусного алгоритма извлечения терминологии

Опорный объект ядра расширяемой онтологии, используемый в дальнейшем анализе, имеет степень близости по отношению к входному слову/сочетанию слов рассчитанную по следующей формуле:

$$k_t = \max_{i=1}^m \left(\frac{n_i}{p_i} \right),$$

где m – число всех объектов ядра OWL-онтологии;

p_i – число слов в лемме опорного объекта ядра OWL-онтологии;

n_i – количество слов из леммы входного словосочетания, найденных в лемме опорного объекта ядра OWL-онтологии.

Общая схема оценки степени близости сочетания слов терминам Про

представлена на рисунке 2.10.



Рисунок 2.10 Тезаурусный критерий. Поиск опорного объекта

При этом порядок следования слов многословия в опорном объекте должен сохраняться.

В случае, если несколько разных объектов онтологии имеют одинаковое значение коэффициента k_t , то опорным будет считаться тот объект, которому соответствует максимальное n_i . Если таких объектов будет несколько, то они все будут считаться опорными и анализ по онтологическому критерию будет проведен для каждого из этих объектов.

Структура онтологии рассматриваемой предметной области предполагает наличие у каждого из ее объектов свойства (DatatypeProperty) «являетсяТермином», имеющее логический тип значения. Это свойство является вспомогательным и определяется экспертом исходя из критерия, насколько данный объект онтологии является характерным конкретно для этой предметной области.

Таким образом, степень терминологичности входного многословия вычисляется следующим образом:

$$k_{Ont} = \frac{k_t}{c+1},$$

где k_t – значение, полученное на первом этапе выполнения алгоритма;

c – число отношений между опорным объектом и ближайшим объектом онтологии, имеющим DataProperty «являетсяТермином»=истина (если данное свойство истинно для опорного объекта, тогда $c=0$). Схема поиска приведена на рисунке 2.11.



Рисунок 2.11 Определение степени терминологичности входных многословий

Если для опорного объекта «являетсяТермином»=ложь, и при этом объект не имеет никаких связей с другими объектами онтологии, либо все связанные объекты также имеют значение «false» этого свойства, то находится другой опорный объект для данного слова/словосочетания и оценка проводится заново. В аналогичной ситуации с другими опорными объектами либо в случае их отсутствия входное одно/многословие признается «не термином» ($k_{Ont}=0$) [105].

Таким образом, процесс оценки степени терминологичности входного многословия представляет собой движение по графу, в узлах которого находятся объекты соответствующих классов онтологий.

2.6.2 Алгоритм вложенных связей

Помимо оценки степени терминологичности отдельно взятого слова/сочетания слов, разработанная метрика позволяет извлечь термины из текста посредством их сопоставления с имеющимися объектами и сочетаниями лемм соответствующих объектов с помощью отношений R_{add} позволяющих расширять набор объектов описываемой предметной области за счет сочетания лемм связанных объектов. Пример: свойства «имеет Отношение» и «является Частью» [48].

Таким образом, при сопоставлении входных сочетаний и объектов

предметной области, связанных между собой однонаправленными отношениями R_{add} , термином рассматриваемой предметной области будет считаться многословие, лемма которой полностью совпадает с объединением лемм соответствующих объектов онтологии.

Метод вложенных связей позволяет извлечь термины из текста посредством их сопоставления с имеющимися объектами и сочетаниями лемм соответствующих объектов с помощью отношений, определенных в онтологии.

Схема разработанного алгоритма представлена на рисунке 2.12.

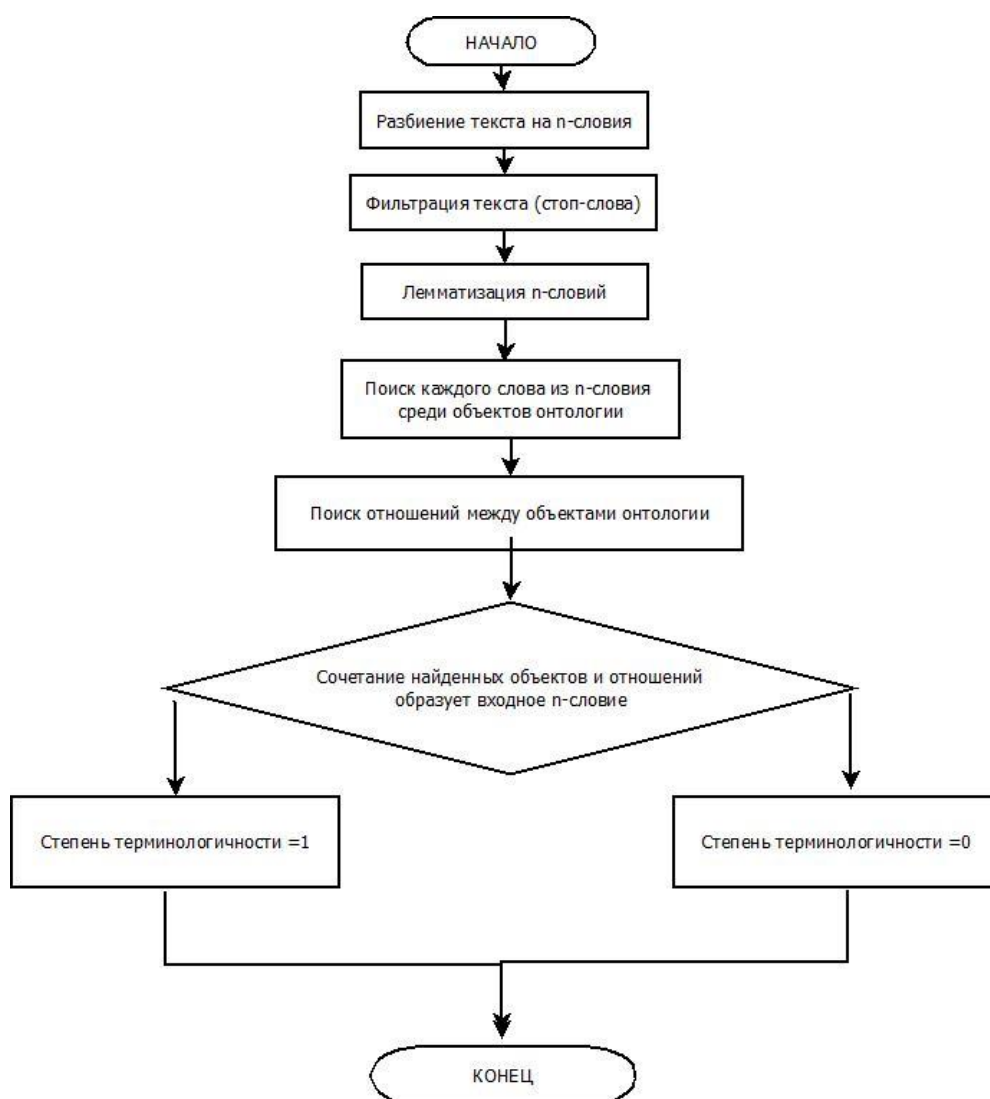


Рисунок 2.12 Схема алгоритма вложенных связей извлечения терминологии из текста

Особенностью данного метода является необходимость представления объектов онтологии преимущественно в виде однословий с максимизацией числа отношений между объектами. Определяющими для использования этого метода являются отношения R_{add} , связь объектов посредством которых позволяет

формировать словосочетания естественным образом.

$$t_1 + R_1 + t_2 + R_2 + \dots + t_i + R_j + \dots + t_m + R_n,$$

где $R_i \in R_{add}$, $t_j \in T$, T — термины прикладной области, которую описывает онтология.

Пример формирования многословий с помощью свойства «имеетОтношение»:

1. Найденная цепочка объектов: «Вращение» + «имеетОтношение» + «Двигатель» + «имеетОтношение» + «ПеременныйТок».
2. Объединение лемм объектов онтологии: «вращение двигатель переменный ток».
3. Термин, извлекаемый из обрабатываемого текста: «вращение двигателя переменного тока».

Пример формирования многословий с помощью свойства «являетсяЧастью»:

1. Найденная цепочка объектов: «Подшипник» + «являетсяЧастью» + «Шпиндель»;
2. Объединение лемм объектов онтологии: «подшипник шпиндель»;
3. Термин, извлекаемый из обрабатываемого текста: «подшипник шпинделя».

Схема данного алгоритма приведена на рис 2.13.

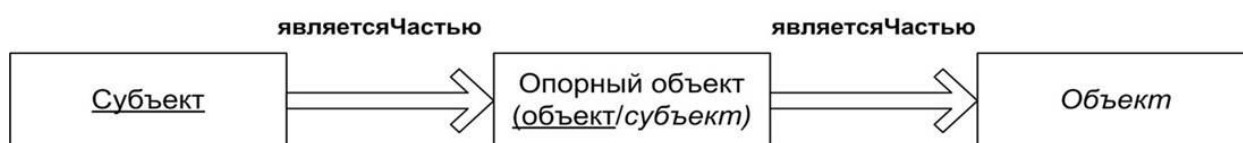


Рисунок 2.13 Алгоритм вложенных связей

При этом извлеченные термины, входящие в свою очередь в термины, состоящие из большего количества слов, не рассматриваются в качестве терминов с целью избегания избыточности [49].

Какой из разработанных критериев является наиболее эффективным в рамках решения задачи расширения ядра онтологии, используемой в качестве базы знаний интеллектуальной САПР, будет рассмотрено и выявлено в главе 4 на

основе проведенных экспериментов.

2.7 Выводы по главе

1 Предложена онтологически-ориентированная методика извлечения терминологии из проектных документов, отличающаяся от известных совместным использованием статистических методов, лингвистических шаблонов и базового ядра предметной онтологии;

2 Разработан алгоритм расширения онтологического представления терминологии предметной области технической системы, отличающийся от известных построением вложенных связей термина-кандидата, извлеченного из проектного документа, до опорного класса базового ядра онтологии;

3 Разработан алгоритм расширения процедурной составляющей базы знаний интеллектуальной САПР, отличающийся от известного использования в качестве прецедента совокупности SWRL-правил и правил решения задачи проектирования;

4 Предложен метод интеграции нечеткого логического вывода и нечеткого онтологического представления терминологии предметной области технической системы, отличающийся от известного использования иерархического нечеткого вывода и модели онтологии класса FuzzyOWL.

Глава 3 Реализация интеллектуальной программной системы логического вывода на основе онтологии для интеллектуальной САПР

В рамках решения задачи оценки эффективности предложенных моделей представления экспертных знаний и алгоритмов обучения и расширения базы знаний на основании анализа архива проектных документов, играющих роль больших корпусов текстов, а также вывода новых знаний, был разработан программный комплекс, состоящий из двух модулей:

1. подсистема онтологически-ориентированного логического вывода на основе интеграции нечеткой онтологии и систем продукционных правил с использованием механизма прецедентов;

2. подсистема расширения ядра онтологии посредством использования алгоритма извлечения терминологии из больших корпусов проектных документов.

Помимо непосредственной программной реализации разработанных алгоритмов и моделей, одной из важнейших задач была разработка необходимой базы знаний в виде нечеткой FuzzyOWL-онтологии, а также заполнение набора SWRL-правил, на основе которых будет делаться логический вывод новых знаний. В связи с этим немаловажным стал процесс выбора наиболее эффективного, удобного, функционального редактора онтологии, который в полной мере является модулем разработанного программного комплекса.

3.1 Выбор инструментов инженерии онтологии объекта проектирования

При решении широкого круга задач, связанных с анализом процессов в сложных системах (задачи проектирования сложных технических систем,

экспертной оценки состояния сложной технической системы и т.д.) использование онтологии проблемной области позволит унифицировать методику анализа и создать определенный смысловой контекст, который обеспечивает восприятие его результатов в удобной для проектировщика форме. Исходя из этого, одной из немаловажных составляющих удачно проведенного анализа является адекватное построение онтологии. В настоящее время для существенного упрощения данной задачи разработано значительное количество программных инструментов, обеспечивающих автоматизацию процесса инженерии онтологий.

Инженерии онтологий можно определить, как совокупность действий, касающихся таких аспектов как:

1. Процесс разработки онтологий;
2. Жизненный цикл онтологий;
3. Методы и методологии построения онтологий;
4. Набор инструментов и языков для их построения и поддержки [104].

Главными программными инструментами разработки предметных онтологий являются редакторы онтологий. Основная функция редактора онтологий – это обеспечение возможности формализации знаний о предметной области в графовой (онтологической) структуре и представление онтологии как спецификации. Основные функции редакторов онтологий:

- создание онтологии в интерактивном режиме;
- работа с элементами онтологии: создание, редактирование, удаление классов, объектов, отношений, аксиом онтологии.
- экспорт/импорт онтологии в нужный формат/из внешних форматов;
- редактирование метаданных онтологии (общего описания, пространства имен, версии формализации и др.).

Помимо этого, большинство редакторов поддерживает расширение функционала посредством подключения дополнительных модулей, плагинов, обеспечивающих проверку логической целостности и непротиворечивости, визуализацию онтологии и т.д.

В настоящее время существует более шести десятков редакторов онтологий, различающихся своими функциональными возможностями, форматами представления данных, доступностью исходного кода и т.д. Сравнительная характеристика наиболее популярных редакторов онтологий приведена в таблице 3.1 [138].

Таблица 3.1 Сравнительная характеристика современных редакторов онтологии

Редактор онтологии	Язык представления	Модель	Язык ПО	Хранение онтологий	Расширяемость
Protege	OKBC	local	Java	Файлы,СУБД	Плагины
Ontolingua	Ontolingua	web	Lisp	Файлы	-
OntoStudio	OXML	local	Java	Файлы, СУБД	Плагины
OntoEdit	OXML	local	Java	Файлы	Плагины
OilEd	DAML+OIL	local	Java	Файлы	-
OntoSaurus	LOOM	web	Lisp	Файлы	-
WebOnto	OCML	web	Lisp, Java	Файлы	-
WebODE	-	web	Java	СУБД	Сервер приложения

Большинство редакторов онтологий позволяет подключать блоки рассуждений (англ. «reasoner» или рус. ризонер), представляющие собой отдельные программные системы (машины вывода), и с их помощью производить логический анализ онтологии, а также выводить запрашиваемые данные из обрабатываемой онтологии. Сравнительная характеристика наиболее известных и часто используемых ризонеров приведена в таблице 3.2 [103].

Таблица 3.2 Современные машины вывода (ризонеры)

Ризонер	Доступность	Язык	Интерфейс
Celebra Engine	commercial	C++	OWL-API
FACT++	free	C++	OWL-API, DIG, lisp-interface
HermiT	free	Java	OWL-API
KAON2	free	Java	Java-based, DIG

Pellet	free	Java	OWL-API, DIG, Jena
RacerPro	commercial	Lisp	OWL-API, DIG
SHER	commercial	Java	OWL-API, DIG, Jena
TrOWL	free	Java	OWL-API, Jena
CEL	free	Lisp	DIG

Наиболее распространенным редактором онтологий является редактор Protégé - он свободно распространяемый, открытый и имеет легко расширяемую архитектуру за счет поддержки модулей расширения функциональности.

Учитывая все преимущества, в рамках данного исследования был использован для построения, анализа непротиворечивости, а также визуального представления OWL-онтологии редактор Protege 4.2 [31]. Помимо встроенного функционала, указанный редактор был расширен следующими свободно распространяемыми модулями:

- OWLViz, VOWL – набор java-библиотек, обеспечивающих визуализацию иерархии классов, свойств и объектов OWL-онтологии;
- FuzzyOWL Plugin – набор библиотек и классов, предоставляющих возможность создания и представления нечетких элементов онтологии в нотации FuzzyOWL.

Посредством описанных инструментов, в рамках данного исследования было разработано 3 предметных онтологии, которые выступали в качестве базы знаний для проведения вычислительных экспериментов с целью оценки эффективности разработанных моделей и алгоритмов.

3.2 Диаграмма вариантов использования разработанной системы

Диаграмма вариантов использования (Use Case Diagram) описывает функциональное назначение системы. Она является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки. Для разработанного программного комплекса диаграмма вариантов использования представлена на рисунке 3.1

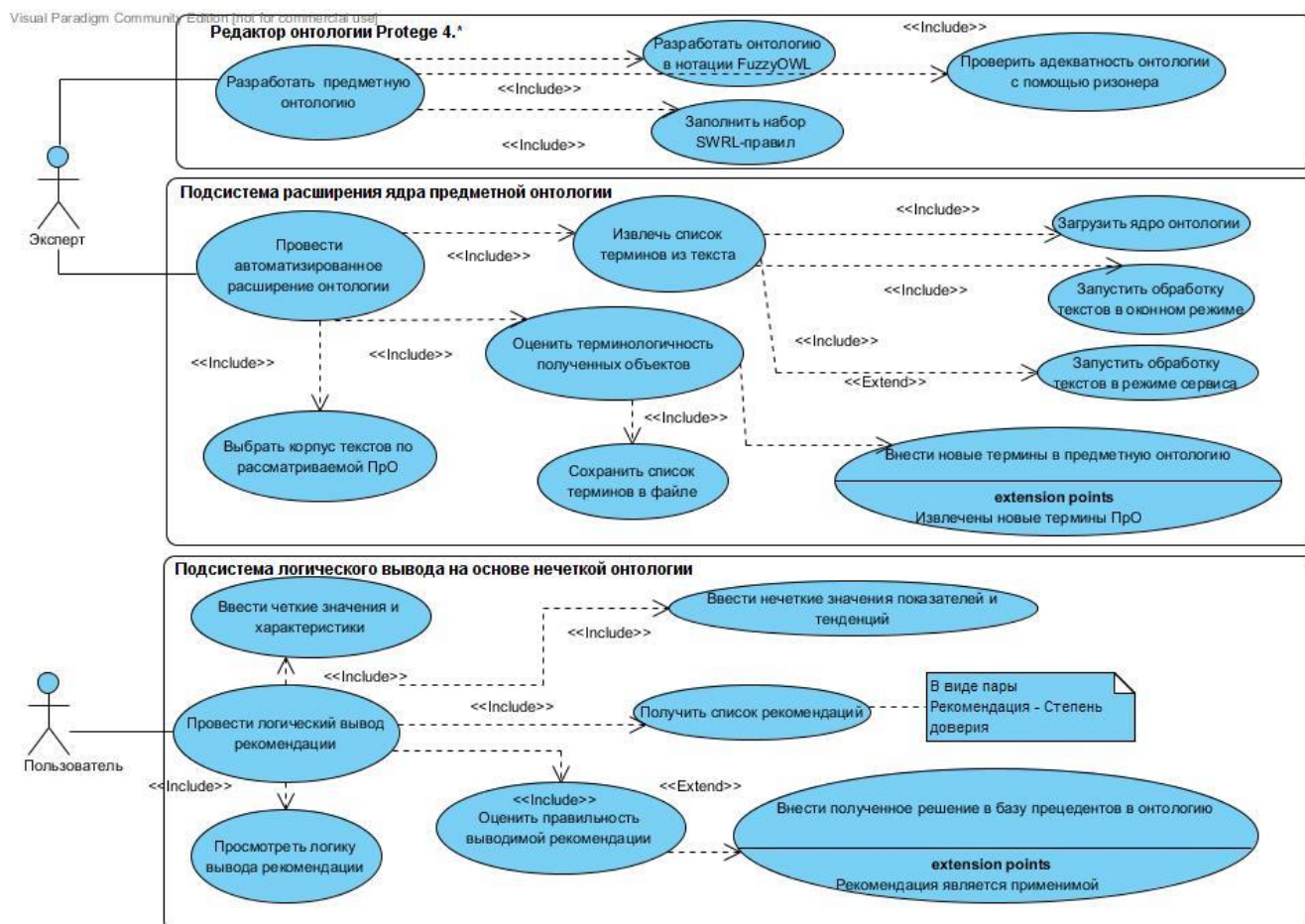


Рисунок 3.1 Диаграмма вариантов использования разработанного программного комплекса

Как видно из схемы, работать с проектируемой системой могут два типа пользователей в зависимости от решаемых задач:

1. Эксперт, который работает с двумя модулями системы:

- редактором онтологии (Protégé 4.x), посредством которого накопленные экспертные знания в данной предметной области формализуются в виде нечеткой онтологии и набора правил, посредством которых далее будет реализован вывод необходимой рекомендации Пользователю. Помимо этого, эксперт имеет возможность оценить адекватность разработанной онтологии посредством использования встроенного ризонера (в Protégé 4.x машины вывода – HermiT и FACT++) в редактор онтологии.
- подсистемой расширения ядра онтологии, которая позволяет извлекать терминологию из обрабатываемых текстов конкретной предметной области, а эксперт имеет возможность оценивать степень терминологичности извлекаемых сочетаний и вносить их в онтологию, таким образом расширяя и обучая базу

знаний.

2. Пользователь, т.е. проектировщик, которому необходимо принять проектное решение, решить задачу или оценить состояние проектируемой системы с использованием накопленных экспертных знаний. Для этого проектировщику предоставляется доступ к подсистеме онтологически-ориентированного логического вывода на основе нечеткой онтологии, которая является полноценной обучаемой интеллектуальной компонентой САПР, так как при верном выводе рекомендации пользователь имеет возможность внести данный вариант в базу знаний в качестве прецедента.

3.3 Подсистема онтологически-ориентированного логического вывода на основе нечеткой онтологии

В рамках решения задачи сравнения алгоритмов логического вывода на основе четкой и нечеткой онтологий была разработана программная система TSAnalyzer 2.1 [116], а также модифицирована OWL - онтология анализа состояния ЛВС согласно методике FuzzyOWL.

Основные функции онтологически-ориентированной системы логического вывода на основе нечеткой онтологии:

- хранение представления особенностей предметной области в виде предметной онтологии;
- логический вывод экспертных рекомендаций в порядке их релевантности на основе результатов анализа входных значений параметров исследуемой системы и обработанной онтологии.

Архитектура разработанной системы представлена на рисунке 3.2.

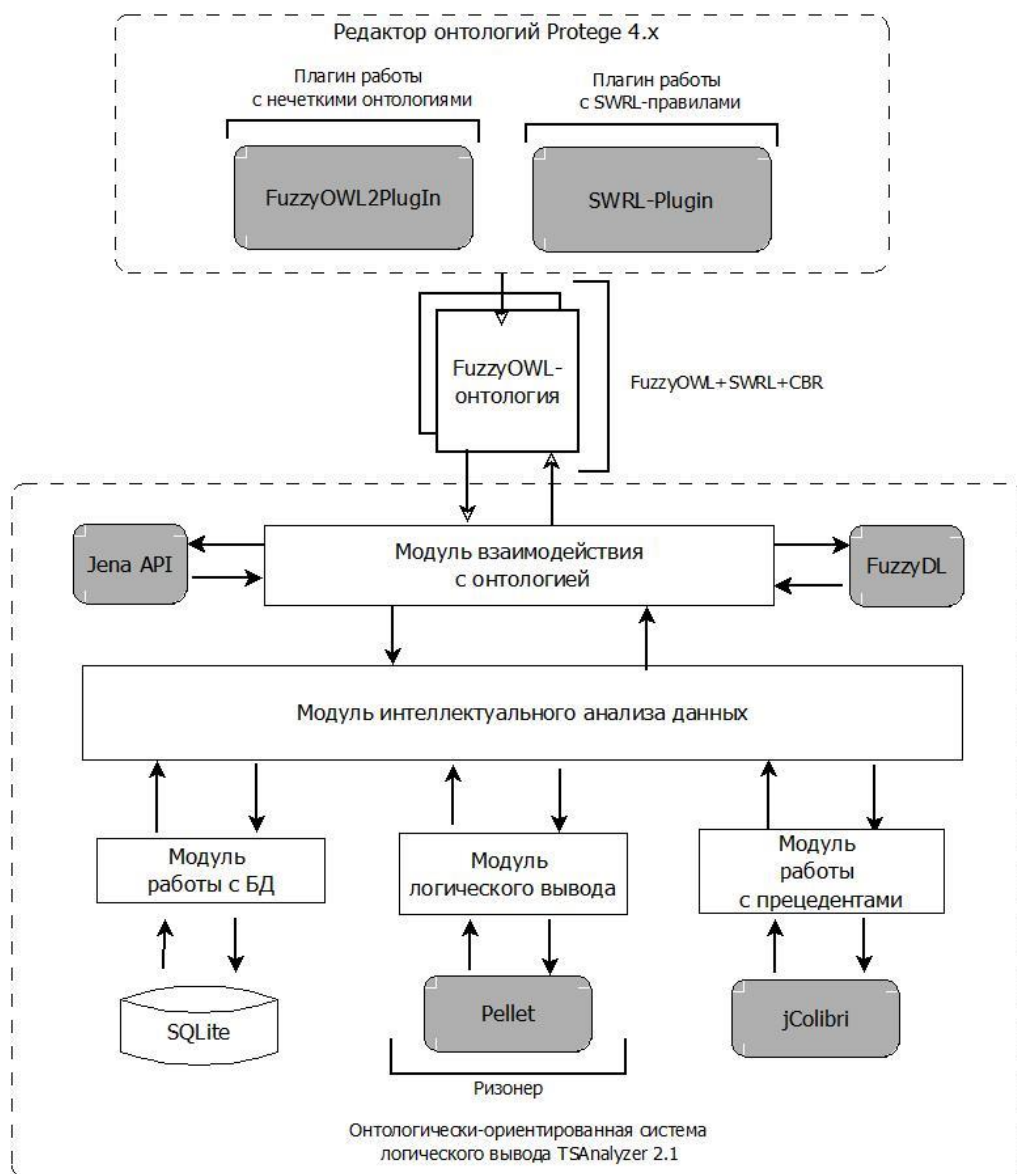


Рисунок 3.2 Архитектура онтологически-ориентированной системы логического вывода

3.3.1 Описание проекта интеллектуальной компоненты САПР

Модуль интеллектуальной компоненты САПР, отвечающий за логический вывод знаний на основе предметной онтологии, был реализован на языке программирования Java в среде разработки Eclipse Luna. Одно из преимуществ языка Java – кроссплатформенность, именно поэтому данная система может работать на вычислительных машинах с ОС MS Windows, GNU/Linux, *BSD и Solaris [107, 77].

При разработке подсистемы были использованы следующие программные компоненты:

- Swing – фреймворк для разработки графического интерфейса системы.

- Apache Jena – набор библиотек и классов, обеспечивающих создание программной среды для работы с онтологиями в формате RDF, RDFS и OWL, а также поддерживающих возможность формирования запросов к онтологии на языке SPARQL.

- FuzzyDL – набор библиотек и классов для работы нечеткими онтологиями формата FuzzyOWL.

- SQLite-jdbc – набор библиотеки классов для работы с СУБД SQLite.

- Pellet – набор библиотек, позволяющий обрабатывать знания в OWL/SWRL-формате, основанном на соответствующей дескриптивной логике (ризонер).

- jColibri – набор библиотек, реализующий механизм прецедентов [128, 129].

3.3.2 Диаграмма классов

На рисунке 3.3 представлена диаграмма классов разработанной подсистемы онтологически-ориентированного логического вывода.

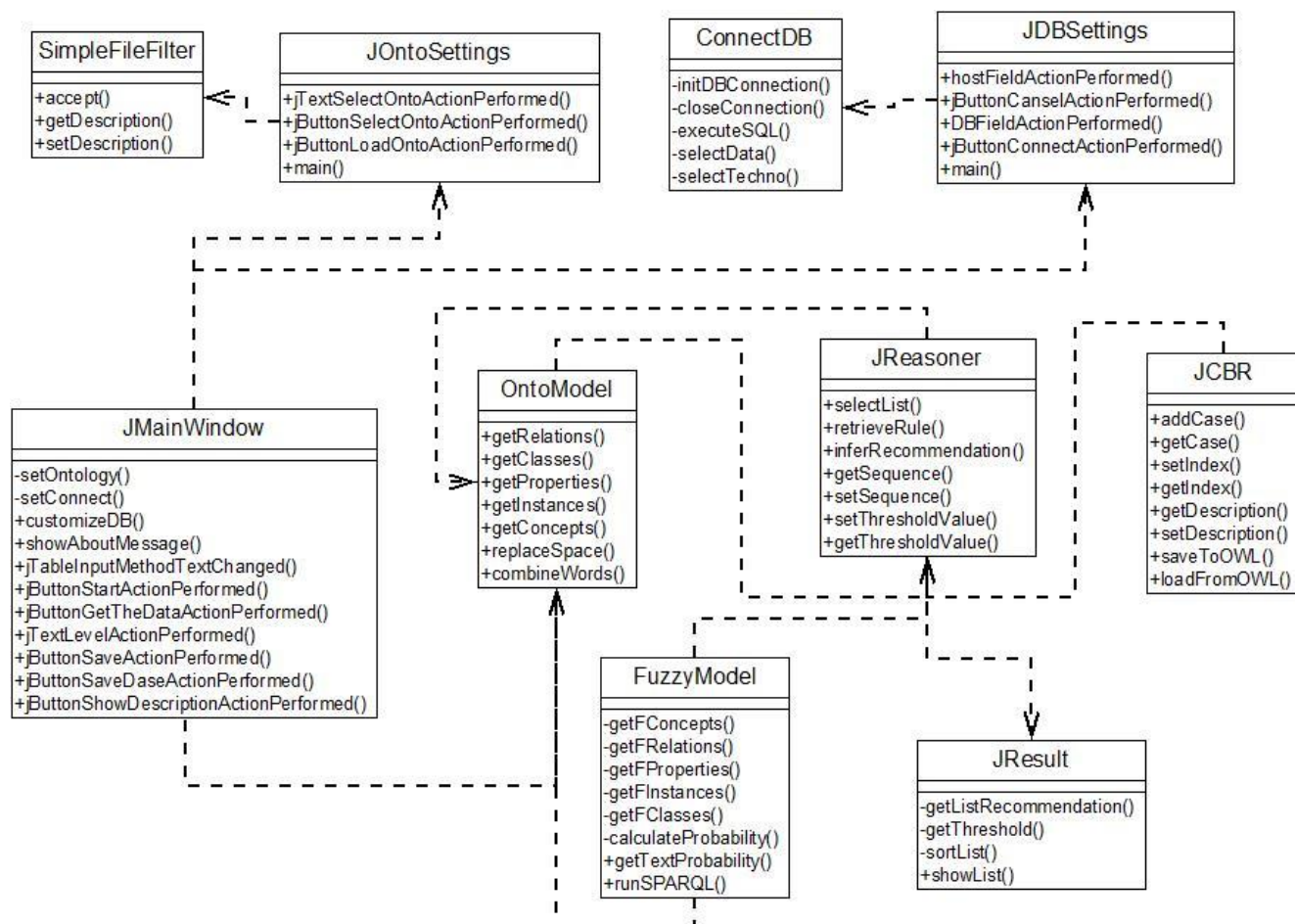


Рисунок 3.3 Диаграмма классов онтологически-ориентированной системы логического вывода TSAlyzer 2.1

Описание основных классов программной системы:

- **ConnectDB** – класс, обеспечивающий работу с реляционной БД для хранения вспомогательных данных, а также в случае получения входных значений системы из результатов анализа данных предметной области.
- **OntoModel** — класс, реализующий основные методы библиотеки Jena для работы с онтологиями. Импортирует фреймворк Jena.
- **FuzzyModel** — класс, обеспечивающий инструменты работы с нечеткими онтологиями. Импортирует библиотеку FuzzyDL.
- **JReasoner** — класс, обеспечивающий работу с библиотекой Pellet, а также осуществляющий логический вывод на основе онтологии. Импортирует фреймворк Pellet.
- **SimpleFileFilter** — вспомогательный класс, обеспечивающий фильтрацию форматов входных онтологий.
- **JResult** — класс, обеспечивающий вывод рекомендаций в порядке их релевантности.
- **JMainWindow** — класс главного окна системы (форма).
- **JOntoSettings** — класс, обеспечивающий настройку загрузки онтологии в репозиторий (форма).
- **JDBSettings** — класс загрузки данных из БД (форма).
- **JExplanation** – класс подсистемы объяснения решений: позволяет просмотреть последовательность правил, вызванных при выводе выбранной рекомендации.
- **JCBR** — класс работы механизма прецедентов. Импортирует библиотеку jColibri.

3.3.3 Реализация алгоритма интеграции

Система позволяет загружать входные данные из БД, настройки соединения с которой настраиваются в отдельном окне (Settings Database). Благодаря такой

возможности, входными данными системы могут быть показатели системы в форме значений тенденций («Рост», «Спад», «Стабильность») и их характера (тенденция «Рост» имеет значение «Быстрый»).

Для заполнения таблицы данными из БД необходимо нажать на кнопку боковой панели «Get the data». Для очистки основной таблицы есть кнопка «Clear».

Помимо этого, отдельное диалоговое окно отвечает за настройки загрузки онтологии в репозиторий для дальнейшей обработки (Settings Ontology).

Пользовательский интерфейс системы предполагает возможность автоматического извлечения входных данных, представления их в виде RDF-троек и оформления в виде таблицы с соответствующими столбцами Subject – Predicate – Object (см. рис. 3.4).

TSAnalyzer 2.1

File Settings Analysis Help

Initial data analysis

Subject	Predicate	Object
Фоновая нагрузка	имеет тенденцию	рост
Время реакции ПО	имеет тенденцию	рост
Топология сети	имеет значение	звезда
Протяженность домена	имеет размер	небольшой
Коллизии	обнаружены в	начало кадра
Количество станций	превышает	10
Широковещательный трафик	имеет значение проценты	15

Get the data Clear Threshold value 0,5 Start

Recommendations:

Title	Probability (ling.)	Probability (num)	Right
Перегруженность канала	скорее всего	0,8	<input checked="" type="checkbox"/>
Настройки сетевой платы	возможно	0,5	<input type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>

Save in cases Show description

Cases

Case	Right
Перегруженность канала	<input checked="" type="checkbox"/>
	<input type="checkbox"/>

Save

Рисунок 3.4 Главное окно системы TSAlyzer 2.1

Система предполагает наличие фильтрации рекомендаций относительно их степени уверенности, таким образом, проектировщику предоставляются только те рекомендации, степень уверенности которых выше порогового значения. Пороговое значение пользователем устанавливается в поле «Threshold value». Порог может иметь значение от 0 до 1.

При активации какой-либо строки таблицы «Recommendations», в поле «Description» отображается подробное описание данной рекомендации.

Запуск выполнения алгоритма осуществляется нажатием кнопки «Start» правой боковой панели.

Программная система в случае удачно проведенного анализа заполняет таблицу Recommendations:

- Поле «Title» – заголовок рекомендации,
- Поле «Probability (ling.)» – степень уверенности данной рекомендации, выраженная в виде лингвистического значения («Скорее всего», «Возможно» и т.д.). Данное поле зависит от численного значения данного показателя и формируется на основании заранее заданной шкалы.

- Поле «Probability (num.)» – степень уверенности рекомендации в интервале от 0 до 1. Рекомендация не выводится в таблицу в случае, если степень уверенности меньше заданного пользователем порогового значения.

Как и любая экспертная система, разработанный программный комплекс включает в себя модуль объяснения решений. Данный модуль вызывается путем двойного клика на соответствующую строку в таблице «Recommendation». Форма подсистемы объяснения соответствующей рекомендации представлена на рисунке 3.5.

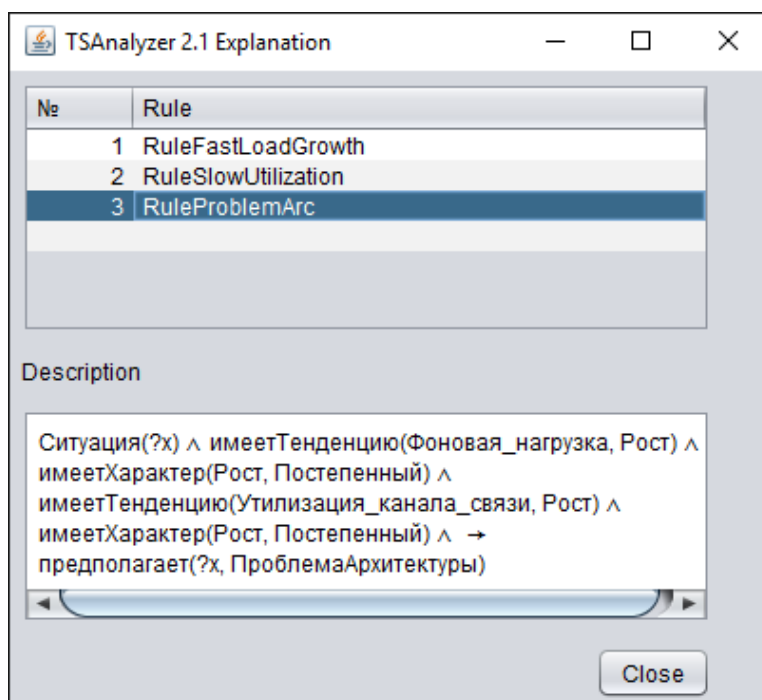


Рис. 3.5. Форма подсистемы объяснения решений

Таблица вызываемой формы содержит последовательность правил, которые были вызваны в процессе логического вывода конкретной рекомендации. В таблице отражаются наименования соответствующих правил, содержание каждого правила можно просмотреть в текстовом поле «Description».

Диаграмма деятельности разработанной системы представлена на рисунке 3.6.

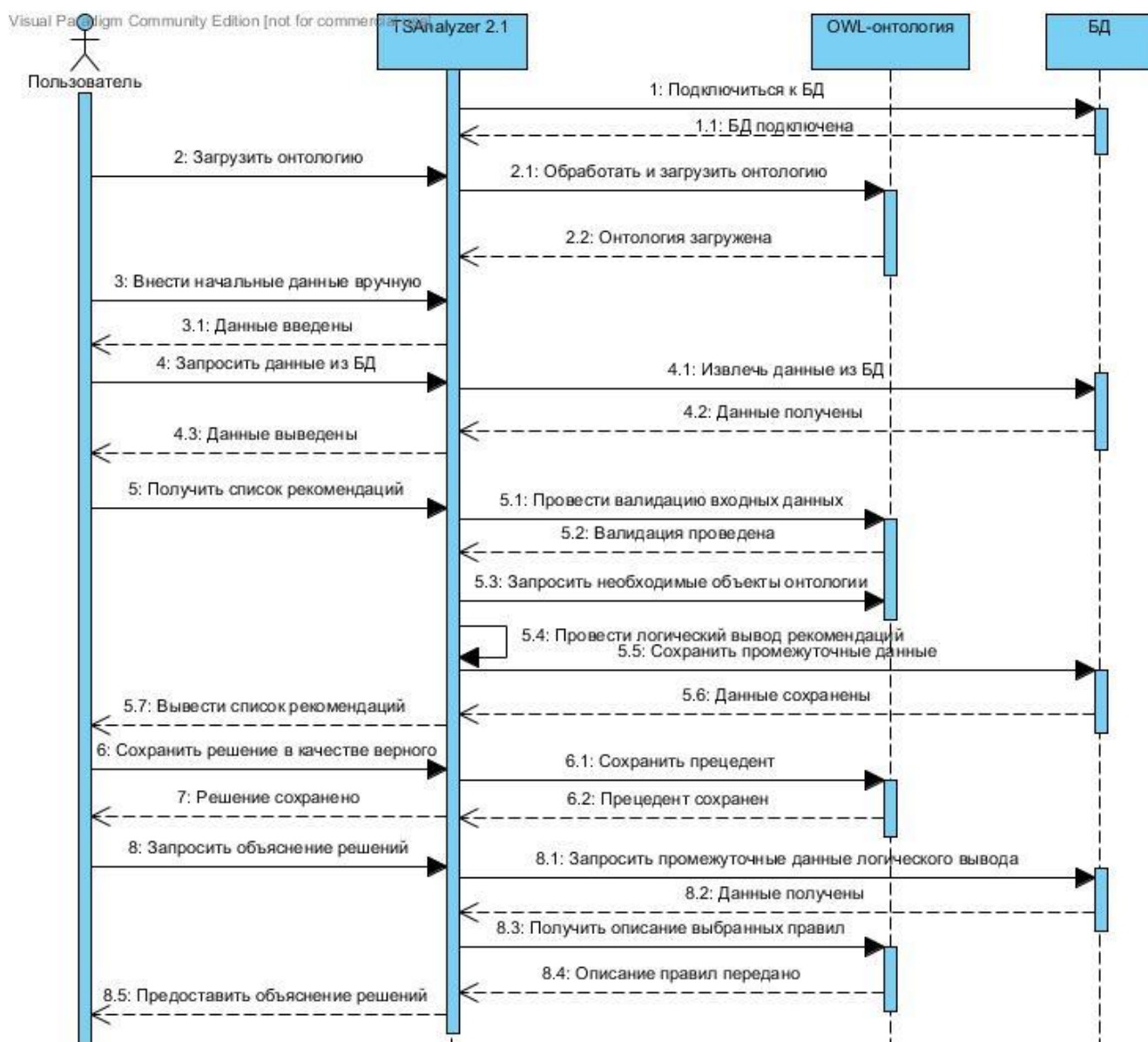


Рисунок 3.6 Диаграмма последовательности системы TSAnalyzer 2.1

Основные этапы работы системы:

1. Получение входных значений параметров анализируемой технической системы. Возможно два способа получения:

- (a) Интерактивный ввод данных через главное окно системы.
- (b) Получение данных из БД, как результата анализа отдельных характеристик предметной области.

Входные значения могут быть следующих видов:

- Численные значения характеристик (например, Количество узлов сети =

14)

- Качественные характеристики (например, ОС сервера = Windows Server 2008 R2)
- Нечеткие значения характеристик системы (Загрузка каналов связи = высокая)
- Нечеткие значения тенденций изменения численных характеристик рассматриваемой системы (Рост количества коллизий = быстрый)

2. Сопоставление входных понятий с объектами онтологии предметной области. Поиск соответствующих объектов, их родителей, свойств и связанных с ними аксиом.

Входными данными системы проектирования и моделирования ЛВС являются:

1. результаты анализа характеристик ЛВС в виде тенденций изменения параметров сети. Результаты хранятся в виде RDF-троек (Например, «Фоновая нагрузка» + «имеет тенденцию» + «рост»). Данные значения хранятся в СУБД SQLite и соответствуют значениям объектов разработанной онтологии;

2. данные, включающие основные структурные характеристики ЛВС, а также данные динамического изменения ее состояния в процессе искусственного повышения загрузки каналов связи. Данные также представлены в виде RDF-троек.

3. Экспертная OWL-онтология, включающая формализованное описание проблемной области, а также набор SWRL-правил логического вывода рекомендаций по устранению проблем в сети.

В процессе логического вывода соответствующей рекомендации система попеременно обращается к онтологии и к набору SWRL-правил, тем самым создавая определенную семантическую среду проведения анализа с привязкой к особенностям проблемной области, описанной в онтологии. Выводимая в главном окне системы рекомендация представляет собой строковое значение отношения «имеетЗначение» соответствующего объекта класса «Рекомендации».

3.4 Подсистема автоматизированного расширения ядра онтологии

3.4.1 Базовый функционал системы

Разработанная подсистема использует файл базы данных base.mdf. На вход подсистеме поступает поток словосочетаний после обработки текста программой Mystem, а также онтология предметной области (Metalworking.owl), созданная с использованием редактора Protégé [47].

Так как процесс извлечения новых объектов предметной области из больших корпусов текстов для пополнения ядра онтологии может занять довольно продолжительное время, разработанная система может быть запущена как в standalone-режиме, так и в режиме сервера, к которому подключается основная программа-клиент. В режиме сервера система осуществляет подсчет разработанных онтологических признаков, принимая на входе n-словие-строку, выдавая на выходе подсчитанное значение терминологичности многословия для занесения его в онтологию.

В standalone-режиме система обеспечивает выполнение следующих функций:

1. обработка (парсинг) ядра расширяемой онтологии проблемной области, выполненной в формате OWL с занесением объектов онтологии в базу данных;
2. обработка свойств объектов, с учетом лемм словоупотреблений;
3. подсчет разработанных онтологических признаков;
4. просмотр содержимого базы данных через встроенную консоль;
5. очистка базы данных.

В качестве особенности системы следует отметить характер обрабатываемых данных. Для корректного извлечения терминов проблемной области из текстов обработка словоупотреблений ведется на уровне их лемм. Таким образом, объект онтологии в базе данных помимо собственного названия на естественном языке содержит еще и его лемму.

Для данного проекта был выбран язык C# (MS VS 2010), на платформе .NET 3.5 и СУБД MS SQL Server 2008. Структура классов разработанной системы

представлена на рис.3.7.

Visual Paradigm Community Edition [not for commercial use]

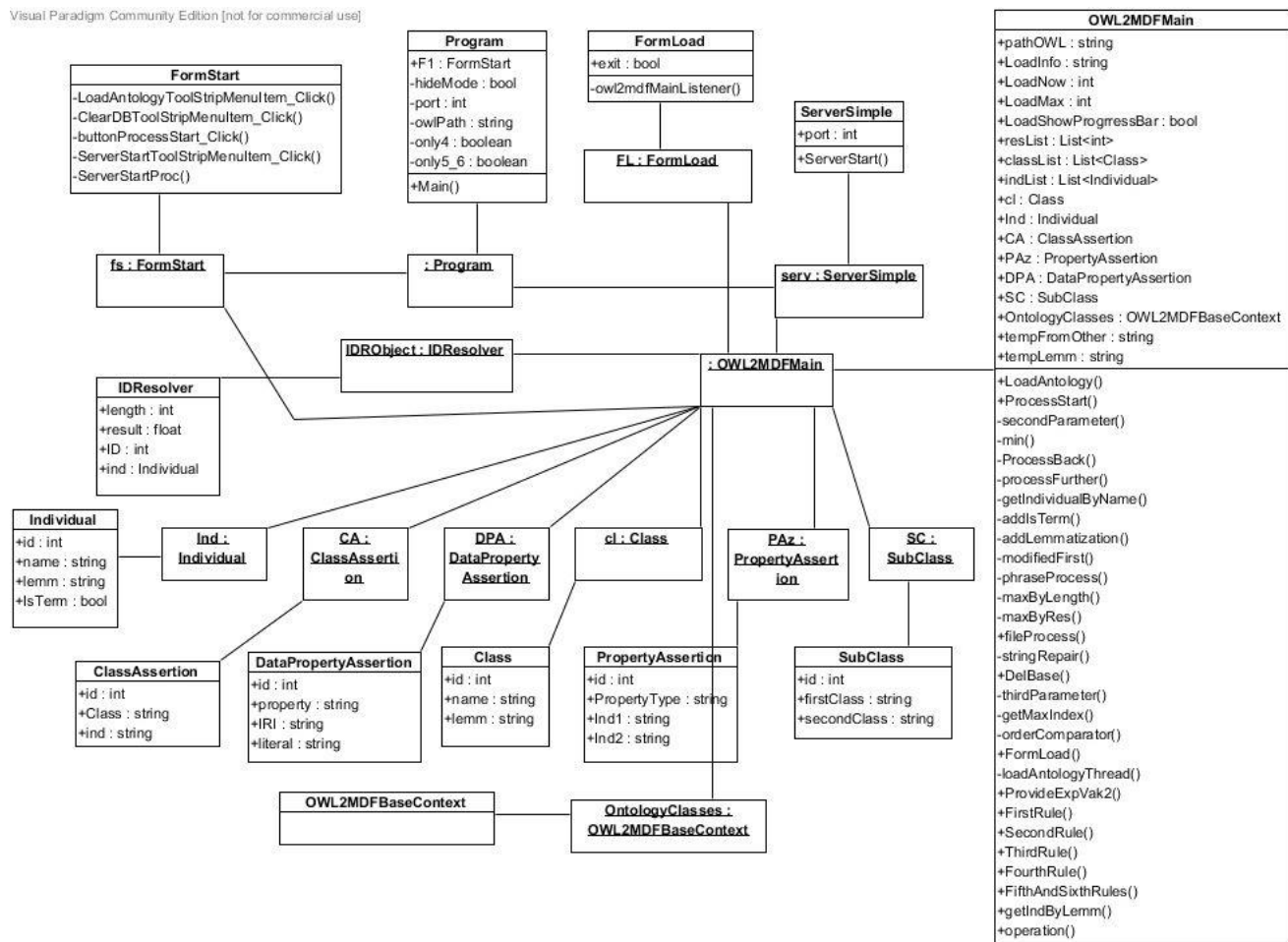


Рисунок 3.7 Диаграмма классов онтологически-ориентированной системы извлечения терминологии

3.4.2 Реализация базы данных

Обращение к таблицам базы данных в данной системе осуществляется с использованием технологии LINQ to SQL. В LINQ to SQL модель данных реляционной базы сопоставляется объектной модели на языке программирования разработчика. При запуске приложения LINQ to SQL преобразует запросы LINQ из объектной модели в SQL и обеспечивает их выполнение в базе данных. Когда база данных возвращает результаты, LINQ to SQL преобразует их обратно в объекты, с которыми возможна работа выбранном языке программирования.

Основное предназначение разработанной базы данных заключается в хранении сущностей выбранной пользователем онтологии и предоставление данных для методов, осуществляющих подсчет критериев. Таблицы базы данных соответствуют сущностям OWL-онтологии, описанным в ранее в диаграмме

классов.

Программа использует БД, которая хранится в двух файлах: с расширением mdf и ldf, которые используются запускаемым в операционной среде экземпляром SQL Server 2008.

3.4.3 Схема информационного обеспечения

Под информационным обеспечением программной системы в данной работе подразумевается совокупность следующих составляющих:

1. Большие корпуса текстов по предметной области, на основе которых проводился экспертный анализ, составлено ядро онтологии «Станки с ЧПУ», расширяемое посредством применения семантических алгоритмов.

2. Программа Mystem, осуществляющая первичную обработку текста и морфологический анализ словоупотреблений. Результат работы Mystem подается на вход разработанной системы.

3. Редактор Protégé, с помощью которой разработано ядро онтологии проблемной области «станки с ЧПУ». Кроме того, редактор позволил исследовать логическую структуру получившейся онтологии, что облегчило разработку и тестирование основной системы

4. Реляционная база данных, составлена для разработанной интеллектуальной компоненты САПР. Для работы системы необходимо упорядоченное хранение выделенных в результате исследования сущностей, а также результатов подсчета семантических признаков.

3.4.4 Диаграмма состояний

Подсистема расширения ядра онтологии работает с несколькими видами информационных ресурсов. Так, тексты из хранилища используются экспертом при работе с Protege; в результате работы эксперта в программе Protégé получается OWL-онтология, которая обрабатывается модулем owl2mdfWindow, заполняя базу данных.

Текст из хранилища, направленный на вход Mystem, автоматически обрабатывается и на выходе предстаёт в виде файла выдачи Mystem. Файл выдачи

обрабатывается в основной программе, заполняя базу данных.

Подсистема расширения онтологии посредством извлечения терминологии может быть запущена в двух режимах: standalone-режиме и режиме сервиса. Диаграмма состояний подсистемы представлена на рисунке 3.8.

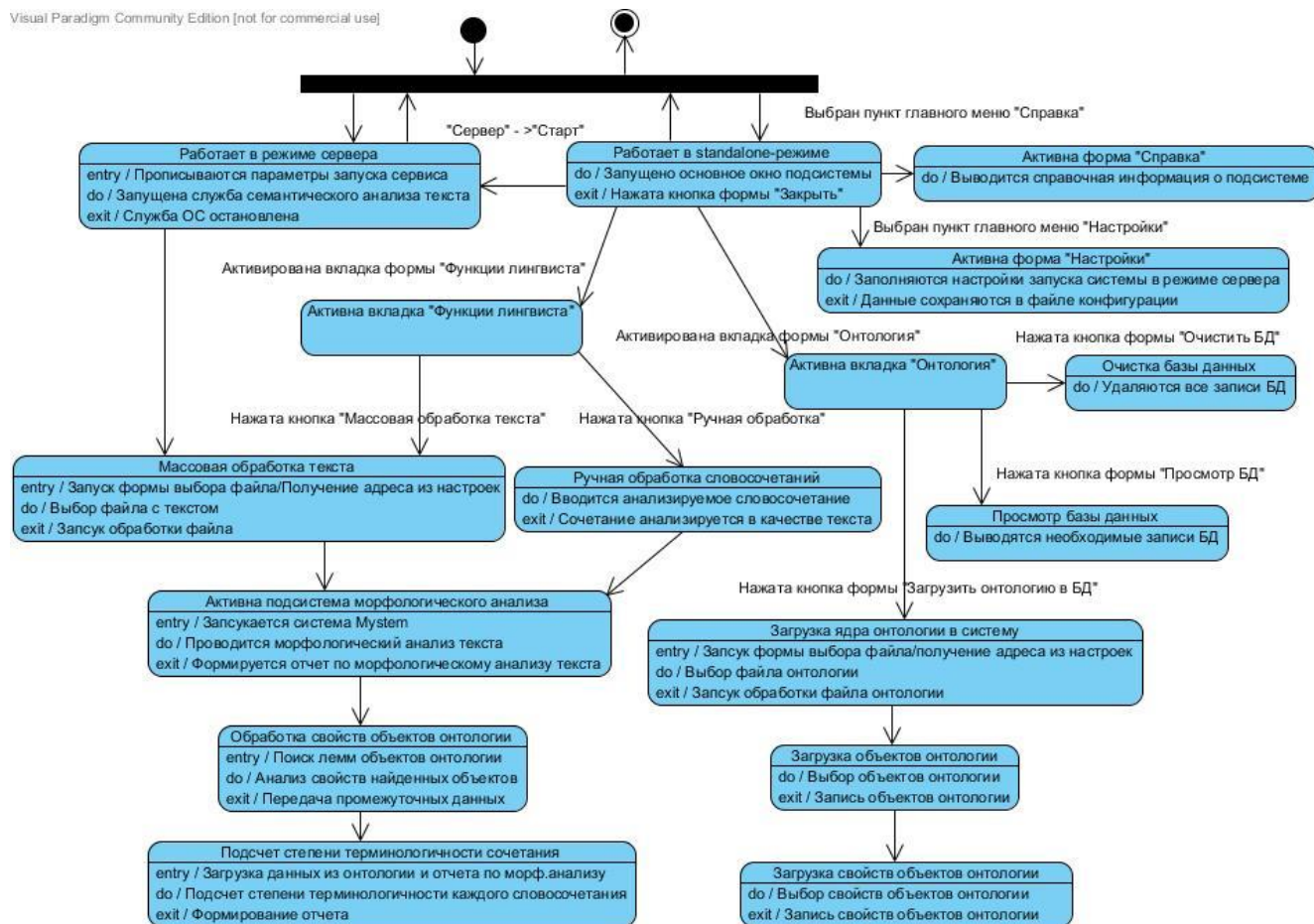


Рисунок 3.8 Диаграмма состояний ООСИТ

Большинство пользовательских функций доступны через оконный интерфейс разработанной системы, открывающийся при запуске исполняемого файла owl2mdfWindow.exe без дополнительных ключей запуска. На рисунке 3.8 представлен внешний вид оконного интерфейса программы.

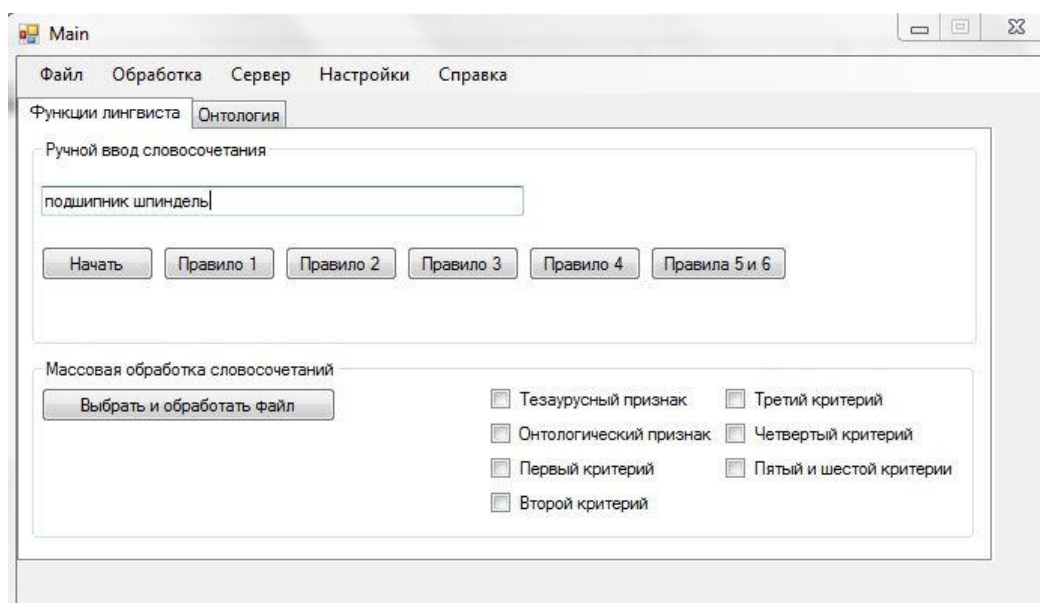


Рисунок 3.8 Внешний вид основной формы ООСИТ

На вкладке «Функции лингвиста» (см. рисунок 3.8) пользовательские элементы управления сгруппированы в две секции: «Ручной ввод словосочетания» и «Массовая обработка словосочетаний».

В секции «Ручной ввод словосочетания» расположено текстовое поле, предназначенное для ввода пользователем слова или словосочетания, по которому будет проверяться то или иное предположение относительно структуры онтологии или для которого необходимо быстро узнать значение какого-либо критерия или значения всех критериев сразу.

Секция «Массовая обработка словосочетаний» содержит кнопку вызова диалога выбора файла со списком слов и словосочетаний, подлежащих обработке, а также ряд флажков, предназначенных для выбора подсчитываемых критериев. Для подсчета значений критериев для слов и словосочетаний из файла на вход программе подается текстовый файл, где каждое отдельное слово или словосочетание расположено на отдельной строке. На выходе программа выдает электронную таблицу формата Microsoft Excel 2007 (*.xlsx) со значениями выбранных критериев для слов и словосочетаний поданного на вход файла.

На вкладке «Онтология» (см. рисунок 3.9) расположены кнопки, предназначенные для выполнения действий по управлению содержимым базы данных, а также список ошибок, появившихся при обработке онтологии.

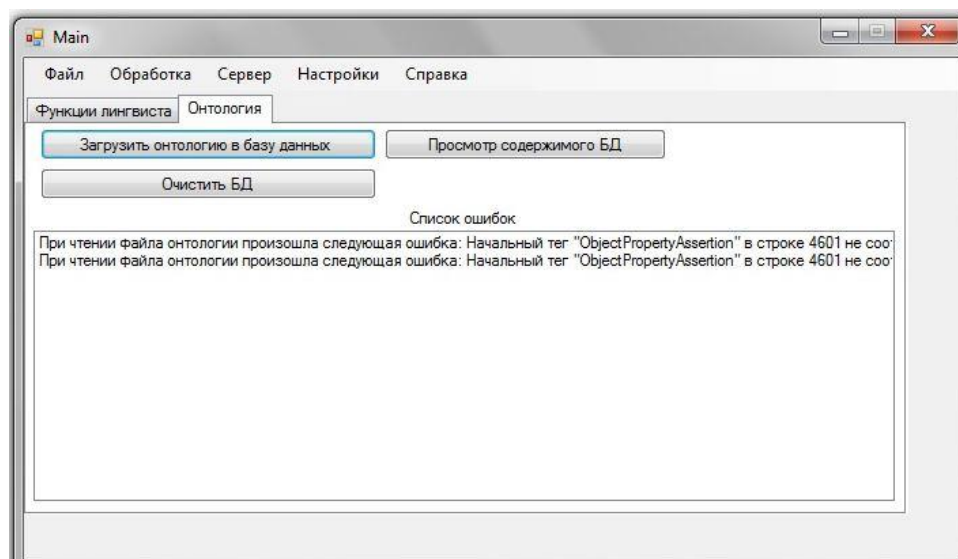


Рисунок 3.9 Вкладка «Онтология» ООСИТ

Процесс обработки онтологии программой сводится к 3-м основным этапам:

1. парсинг файла онтологии с добавлением объектов в базу данных;
2. лемматизация объектов;
3. обработка свойств «являетсяТермином».

В случае обработки большого корпуса текстов по соответствующей предметной области, которая может занять достаточно длительное время, программа была дополнена режимом работы в качестве сервера.

Режим работы в качестве сервера предполагает фоновый запуск программ и прослушивание ею порта 27217 (по умолчанию). Далее начинается обмен информацией с основным модулем системы: получив n-словие, подсистема осуществляет подсчет значений степени терминологичности и затем отправляет результат основному модулю и так далее для каждого n-словия.

3.4.5 Схема работы

Перед запуском программы в режиме сервера необходимо убедиться, что на компьютере не запущен ни один процесс, осуществляющий прослушивание порта 27217 (или другого порта, прослушивание которого планируется), иначе программа не запустится. Реализовано 2 способа запуска программы в режиме сервера.

1. Основной модуль системы автоматизированного извлечения

терминологии из текста автоматически запускает экземпляр программы для подсчета семантических критериев.

2. Экземпляр программы запускается через меню основной формы программы «Сервер-Старт».

Следует привести параметры, которые принимает программа при запуске в режиме сервера:

- server — осуществляет запуск программы в режиме сервера;
- port — задает программе порт для прослушивания, тип данных - целое число;
- path — при запуске в режиме сервера задает путь по умолчанию к обрабатываемой онтологии и осуществляет ее обработку, тип данных - строка.

Закрытие программы, запущенной в режиме сервера, осуществляется через диспетчера задач Windows завершением процесса owl2mdfWindow.exe.

Исполняемый файл программы компилируется из множества исходных файлов. Исходные файлы представляют собой как текст программы на языке C#, так и дополнительные файлы, связанные с базой данных.

Структурная схема исполняемого файла программы представлена следующими составными частями:

- Program.cs — содержит код одноименного класса, является точкой входа приложения;
- ServerSimple.cs — содержит код одноименного класса, который используется при запуске программы в режиме сервера;
- FormStart.cs — содержит код одноименного класса, представляющего собой основную форму программы, используется при запуске программы в режиме оконного приложения;
- owl2mdfMain.cs — содержит код одноименного класса, содержащего методы подсчета критериев метрики и обработки онтологии;
- IDResolver.cs — содержит код одноименного класса, используется в качестве вспомогательного класса при подсчете некоторых критериев;
- OntologyClasses.dbml — содержит схему базы данных и

сгенерированный средой разработки код, позволяющий использовать технологию LINQ to SQL при работе с базой данных;

Структурная схема представлена на рисунке 3.10:

Рисунок 3.10 Структурная схема ООСИТ

Концептуальная схема программной системы, реализующей рассматриваемую онтологическую метрику, включает в себя следующие компоненты:

- два программных модуля, каждый из которых взаимодействует со своей базой данных;
- модуль морфологического анализа Mystem от компании Яндекс;
- корпус проектных документов;
- онтология проблемной области, составленная экспертом в редакторе Protégé (рис. 3.11).

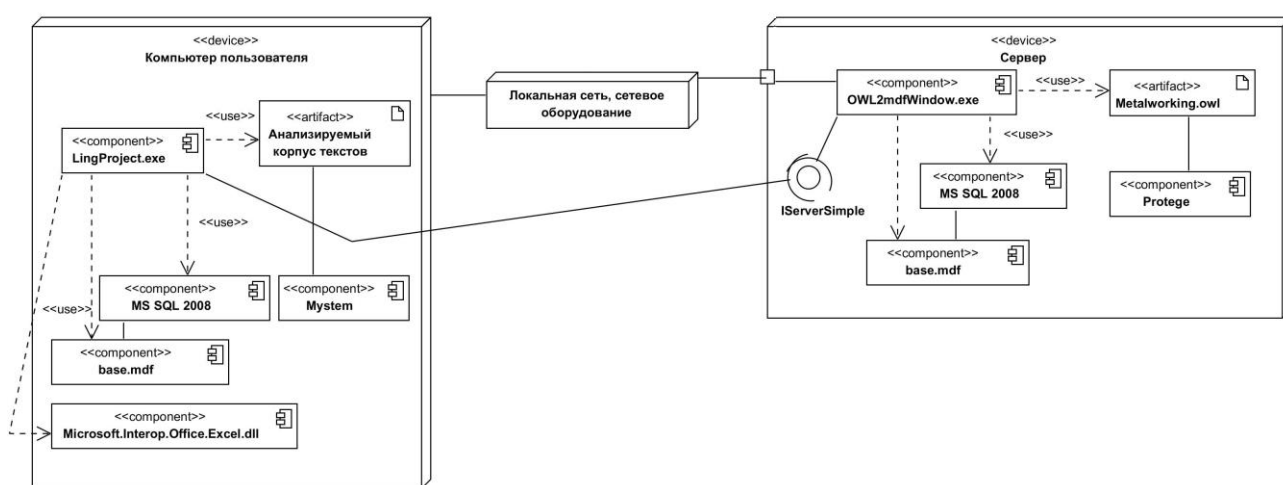


Рисунок 3.11 Диаграмма развертывания онтологически-ориентированной системы извлечения терминологии

Алгоритм работы разработанной системы извлечения терминологии предполагает следующую последовательность действий:

1. Обработка текста модулем морфологического анализа;
2. Подсчет лингвистических и статистических характеристик полученного текста, содержащего морфологическую разметку, основным модулем системы;
3. Подсчет семантических характеристик слов и словосочетаний обрабатываемого текста, базирующийся на представленных методиках с использованием разработанной OWL – онтологии.

3.5 Выводы по главе

- 1 Проведен анализ современных инструментов инженерии онтологий, а

также машин вывода (ризонеров), необходимых для реализации логического вывода. Наиболее универсальным среди наиболее популярных, а также обладающим необходимой функциональностью за счет возможности подключения дополнительных плагинов, оказался редактор Protégé. Среди ризонеров оптимальным вариантом является фреймворк Pellet. Именно эти два приложения использовались при построении онтологически-ориентированной системы логического вывода.

2 Разработана программная система поддержки проектирования, отличающаяся наличием специальной компоненты интеллектуальной САПР для представления терминологии лингвистического обеспечения. Система состоит из двух модулей.

Первый модуль реализует алгоритм интеграции механизмов логического вывода и онтологического представления экспертных знаний с элементами нечеткости, а также использует механизм учета прецедентов с целью расширения ядра онтологии. Модуль позволяет на основании четких и нечетких входных значений характеристик состояния исследуемой технической системы генерировать два набора рекомендаций:

- на основании онтологического представления предметной области (с учетом соответствующих степеней уверенности);
- на основании прошлого опыта экспертов в решении аналогичных задач;

Второй модуль обеспечивает расширение декларативной области OWL-онтологии посредством реализации алгоритма семантического анализа крупных корпусов текстов. Данный модуль реализует два предложенных алгоритма и позволяет обрабатывать большие объемы текстовой информации в фоновом режиме (в виде сервиса).

Глава 4 Анализ адекватности разработанных моделей и методов на основе вычислительных экспериментов и практики применения

4.1 План проведения экспериментов

Онтологии как общий семантический базис сложных итеративных процессов, нуждающихся в глубокой информационно-аналитической поддержке, являются унифицированным средством представления терминологии в ЛО САПР и применимы на различных этапах проектирования СТС.

На схеме 4.1 выделены фрагменты отдельных технологических процессов сборки самолетов для производства тяжелого военно-транспортного самолета Ил-76МД- 90А на АО «Авиастар-СП», к которым в рамках данного исследования, были применены разработанные методы и алгоритмы.

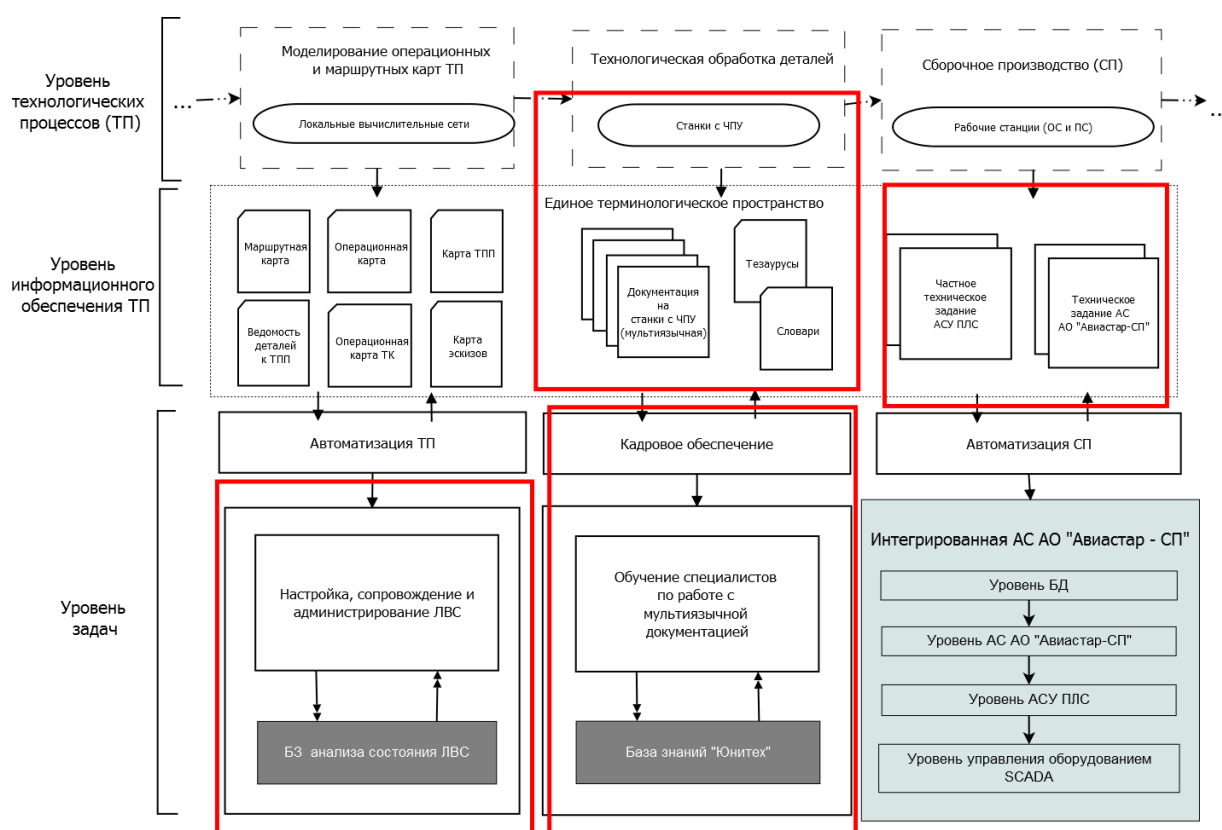


Рисунок 4.1 Фрагмент схемы СТС производственного типа на примере поточной линии сборки самолетов Ил-76МД- 90А на АО «Авиастар-СП»

При решении задачи оценки эффективности разработанных моделей и алгоритмов необходима адаптация условий проведения экспериментов под специфику решаемых задач:

- Использование онтологий особенно эффективно на начальных этапах проектирования сложных систем (при разработке начальной документации на систему, технического задания) для формирования единого терминологического пространства, используемого на последующих этапах проектной деятельности. В связи с этим, для оценки предложенных алгоритмов извлечения терминологии из проектных документов были проведены эксперименты по формированию терминологического окружения процесса разработки частного технического задания АСУ Поточной линии сборки для АО «Авиастар-СП» на основании анализа документации по тематике авиастроения.

- Для оценки алгоритмов интеграции онтологической формы представления знаний, механизма логического вывода с целью поддержки принятия экспертных решений, а также эффективности механизма расширения базы знаний интеллектуальных САПР за счет учета прошлого опыта решения подобных задач, необходим выбор проектной задачи, решение которой предполагает привлечение нескольких экспертов, учет нечеткости в формулировке поставленной задачи, а также рассмотрение возможности ее решения на некотором временном промежутке. Поэтому в качестве объекта экспериментов была выбрана ЛВС в условиях искусственного повышения трафика.

В процессе проведения данных экспериментов были задействованы алгоритмы интеграции нечетких онтологий, системы логического вывода на основе использования SWRL-конструкций, а также был задействован механизм прецедентов с целью расширения сформированной базы знаний.

- Для оценки эффективности предложенных алгоритмов расширения ядра онтологии интеллектуальной САПР посредством извлечения терминологии из проектной документации был проведен ряд экспериментов по формированию базы экспертных знаний «Юнитех» одноименного учебного центра с

использованием готового семантического ядра в виде тезауруса, составленного на базе руководства по эксплуатации токарно-фрезерного станка с числовым программным управлением (ЧПУ).

4.2 Эксперименты по интеграции онтологического анализа и логического вывода в задачах автоматизированного проектирования

Для проверки адекватности алгоритма интеграции онтологической и продукционной моделей представления экспертных знаний, а также корректности работы интеллектуальной САПР, реализующей данный алгоритм, был проведен ряд экспериментов, в рамках которых были смоделированы возможные проблемные ситуации, возникающие в процессе работы ЛВС при искусственном повышении загрузки каналов связи [137].

В рамках эксперимента были проведены следующие действия:

1. Группой экспертов была разработана нечеткая FuzzyOWL-онтология с включением набора SWRL-правил.

2. Далее на тестируемой ЛВС, технические характеристики которой изложены в таблице результатов экспериментов, было смоделировано 30 проблемных ситуаций снижения производительности ЛВС, причины появления которых необходимо было распознать.

3. Показатели оборудования в каждой моделируемой ситуации, а также параметры сети и архитектуры ЛВС заносятся в систему TSAalyzer 2.1, которая в качестве базы знаний использовала разработанную группой экспертов на 1-ом этапе предметную онтологию. На основании введенных данных и хранящихся знаний, АС производила логический вывод одной или нескольких рекомендаций по корректировке состояния сети.

4. В целях проверки эффективности разработанных алгоритмов правильность полученных рекомендаций оценивались группой экспертов, не задействованных при разработке данной онтологии. В случае вывода верного решения оно заносится пользователем в базу прецедентов онтологии с помощью

реализованного функционала системы TSAnalyzer 2.1.

4.2.1 Онтология ЛВС

В рамках решаемой задачи онтологического анализа состояния ЛВС были проведены следующие действия:

1. Экспертом в области проектирования ЛВС построена FuzzyOWL-онтология соответствующей проблемной области [115];
2. Был составлен набор продукционных правил в формате SWRL, включенных в файл онтологии проблемной области;
3. Была разработана онтологически-ориентированная система логического вывода с элементами нечеткой логики.

Разработанная OWL-онтология анализа состояния ЛВС имеет иерархическую организацию и включает в себя 81 класс, 104 свойства классов и порядка 200 экземпляров классов онтологии. Онтология имеет 6 уровней иерархии, что позволяет максимально конкретизировать термины предметной области, используемой при решении поставленной задачи.

Фрагмент онтологии (первые 3 уровня иерархии классов) анализа состояния ЛВС представлен на рисунке 4.1.

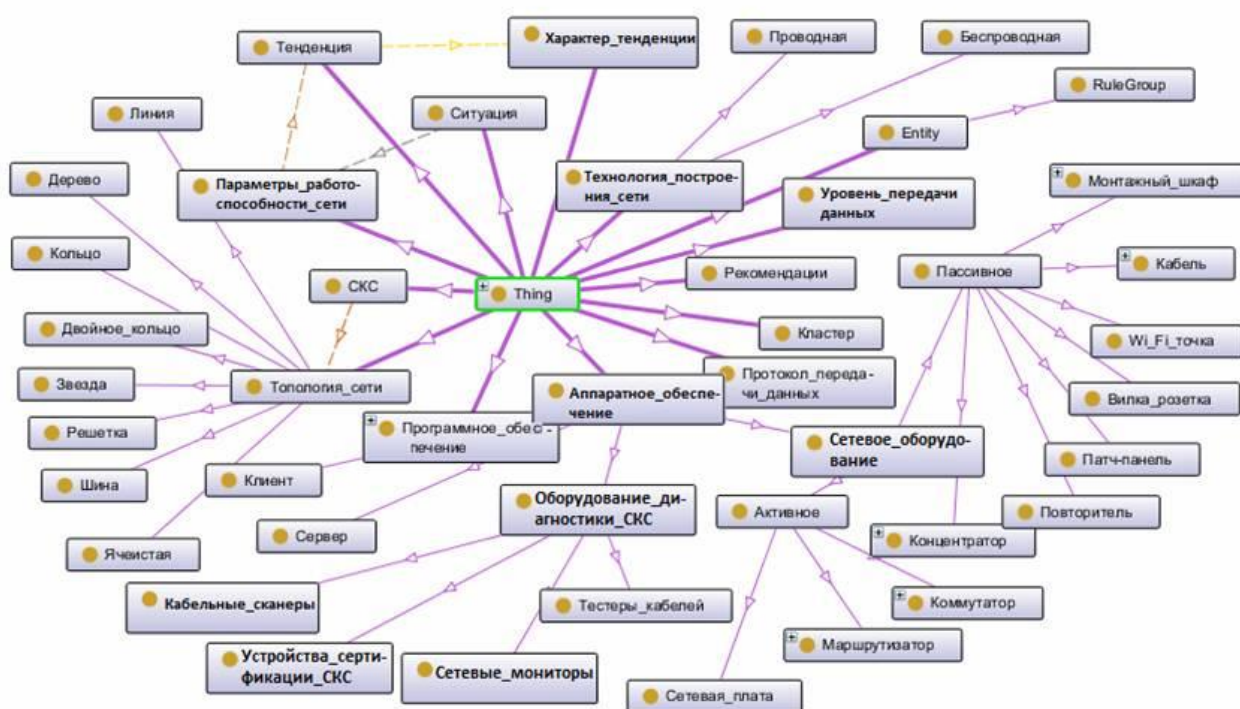


Рисунок 4.1 Фрагмент онтологии анализа состояния ЛВС

Пример объявления класса «Анализаторы протоколов» разработанной OWL-онтологии:

```
<owl:Class rdf:ID="Анализаторы_протоколов">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#ПО_мониторинга_и_анализа_сетей"/>
  </rdfs:subClassOf>
</owl:Class>
```

Пример объявления экземпляра класса «Параметры работоспособности сети»:

```
<Параметры_работоспособности_сети
rdf:ID="Интенсивность_генерируемой_нагрузки">
  <rdfs:label rdf:resource="#Интенсивность_генерируемой_нагрузки"/>
</Параметры_работоспособности_сети>
```

Возможность адекватного представления объектов и отношений рассматриваемой предметной области достигается в процессе построения онтологии в редакторе Protégé благодаря специальным интерактивным инструментам, задающим конкретные характеристики отдельным отношениям.

В связи с тем, что любая OWL-онтология строго иерархична и не предполагает наличия встроенных отношений типа «часть – целое», в разработанной онтологии данный недостаток восполняют взаимнообратные свойства «являетсяЧастью» и «включает».

Пример отношения типа данных «имеетВнутреннююПропускнуюСпособность» объектов класса «Коммутатор»:

```
<owl:DatatypeProperty rdf:ID="имеетВнутреннююПропускнуюСпособность">
  <rdfs:domain rdf:resource="#Коммутатор"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
```

Пример отношения свойства «имеетТопологию» объектов класса «Структурированная кабельная система»:

```
<owl:ObjectProperty rdf:ID="имеетТопологию">
```

```

<rdfs:domain rdf:resource="#CKC"/>
<rdfs:range rdf:resource="#Топология_сетей"/>
</owl:ObjectProperty>

```

Одним из преимуществ онтологий в формате OWL DL является возможность проверки их логической целостности и непротиворечивости посредством применения машины вывода (ризонера). Одна из основных услуг, предлагаемых ризонером, – тестирование иерархии классов и их описаний. В используемом в процессе создания онтологии редакторе Protégé 4.2 имеются встроенные машины вывода Fact++ и HermiT, запуск работы которых позволил избежать следующих ошибок структуры данных внутри разработанной онтологии:

1. Ошибки согласованности классов: класс считается несогласованным, если он не может иметь экземпляров, вследствие множественного наследования от непересекающихся классов.

2. Ошибки согласованности свойств экземпляров классов: заключаются в противоречивости установленных пользователем характеристик свойства (свойство r является одновременно функциональным и обратно функциональным по отношению к заданным множествам классов домена и диапазона).

Разработанная онтология анализа состояния ЛВС хранится совместно с набором SWRL-правил, которые играют роль драйвера в процессе логического вывода и непосредственно связаны с предметной областью, описанной в онтологии.

Архитектура ЛВС, некоторые моделируемые проблемы в процессе искусственного повышения трафика в сети и семантический вывод системы TSAAnalyzer, содержащий рекомендацию по решению проблем, представлены в таблице 4.1.

Таблица 4.1 Моделируемые проблемы в процессе искусственного повышения трафика

Характеристики ЛВС	Моделируемые проблемы	Вывод системы TSAAnalyzer
-----------------------	--------------------------	---------------------------

<p>Топология – «звезда», количество рабочих станций 14, ОС сервера – Windows Server 2008 R2, сетевой коммутатор D-Link DGS-3420-28SC , 20 портов, объем оперативной памяти коммутатора– 2 Mb, режим работы 100 Мб/с Full Duplex, режим борьбы с коллизиями –CSMA/CD, QoS вкл, тип коммутационных шнуров – UTP, категория – 5Е, разъемы RJ-45.</p>	<p>Быстрый рост фоновой нагрузки, рост времени реакции пользовательского ПО</p>	<p>Причина – перегруженность канала связи. Требуется изменение архитектуры сети. Число станций в перегруженных доменах следует уменьшить, а станции, создающие наибольшую загрузку, – подключить к выделенным портам коммутатора</p>
	<p>Активация режима Half Duplex, рост фоновой</p>	<p>Причина – несоответствие характеристик сетевого коммутатора объему трафика. Требуется замена коммутатора или изменение его настроек</p>
	<p>Повреждение одного из коммутационных шнуров, рост числа коллизий</p>	<p>Причина – неполадки с подключением коммутационных шнуров. Причина – неправильная организация заземления компьютеров, включенных в локальную сеть</p>
	<p>Наличие дефектов у портов коммутатора, рост числа ошибок CRC, рост фоновой нагрузки</p>	<p>Причина – неполадки работы портов сетевого коммутатора или повреждение коммутационного шнура, ведущего к проблемному хосту</p>
	<p>Наличие дефектов сетевой платы, быстрый рост числа локальных и удаленных</p>	<p>Причина – проблема в настройках сетевой платы сервера домена, проверьте целостность сетевой карты и правильность ее настройки</p>

Стоит отметить, что дальнейшая детализация правил вывода с целью получения более точного и адекватного значения рекомендации может быть реализована по двум сценариям:

1. За счет увеличения количества атомов в антецеденте правил. Таким образом, каждое правило будет охватывать большее число параметров, от которых может зависеть производительность сети. Главным недостатком такого пути является заметное замедление процесса обработки модифицированных правил.

2. За счет разбиения правил на более короткие с увеличением количества SWRL-конструкций, следствием которых будет присвоение конкретного значения отдельному параметру сети с его последующим участием в процессе логического вывода в качестве входных данных другого правила. Данный подход является рациональным с точки зрения временных затрат и вариативности исходов процесса вывода.

4.3 Сравнение методик логического вывода на основе четких и нечетких онтологий

С целью сравнения предложенной методики взаимодействия онтологического анализа и механизмов логического вывода на основе четкой и нечеткой онтологии был проведен ряд экспериментов, в рамках которых были смоделированы возможные проблемные ситуации, возникающие в процессе работы локальной вычислительной сети при искусственном повышении загрузки каналов связи [141].

Сравнительная характеристика методик логического вывода в процессе моделирования проблемных ситуаций работы ЛВС представлена в таблице 4.2 и на рис.4.2.

Таблица 4.2 Сравнение методик логического вывода на основе четких (OWL+SWRL) и нечетких (Fuzzy OWL+SWRL) онтологий

Рекомендации	OWL+SWRL	FuzzyOWL+SWRL
Нет решений	3	1
Несколько решений (в том числе верное или близкое к верному)	3	25
Несколько неверных решений	-	-
Одно верное (близкое к верному) решение	23	4
Одно неверное решение	1	-

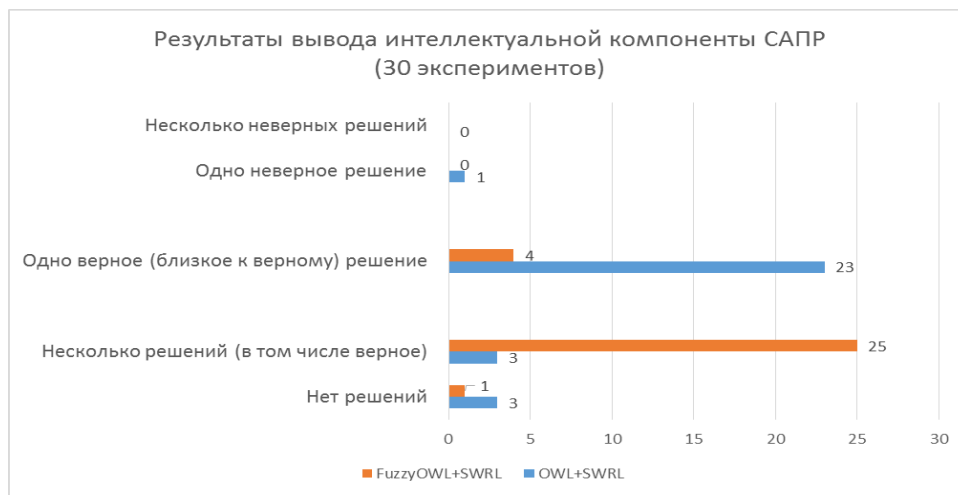


Рисунок 4.2 Результаты логического вывода рекомендаций интеллектуальной компоненты САПР

Полная таблица результатов проведенных экспериментов по логическому выводу рекомендации на основе интеграции нечеткой онтологии и набора SWRL-правил представлена в Приложении Д.

Как видно из результатов проведенных экспериментов, благодаря включению нечеткости в описание предметной области, значительно снижаются риски потери возможных результатов работы блока логического вывода, увеличивается гибкость процесса вывода, а пользователю предоставляется гораздо широкий выбор вариантов, которые при этом упорядочены по мере их релевантности с точки зрения совокупности экспертных знаний, заложенных в онтологии и наборе правил.

4.4 Проведение экспериментов по логическому выводу с элементами интерактивного обучения

Таблица 4.3 Сравнение методик логического вывода знаний (FuzzyOWL+SWRL и CBR+ FuzzyOWL)

Номер/ (Партия экспериментов)	Количество экспериментов			Количество использованных прецедентов из базы
	Результат эксперимента - нет решений	Результат эксперимента - несколько решений	Результат эксперимента - одно верное решение	
1 (1-10)	1	8	1	1
2 (11-20)	-	9	1	3
3 (21-30)	-	8	2	4

Как видно из результатов, с каждой следующей партией экспериментов вывод правильного решения на основе прецедентов становится чаще, так как система проходит обучение, благодаря внесению верных решений в CBR-базу. Несмотря на то, что включение нечеткости в описание предметной области значительно снижает риски потери возможных результатов работы блока логического вывода и увеличивает гибкость процесса вывода, наличие большого количества вариантов решения задачи не дает пользователю в полной мере положиться на какой-либо из предложенных вариантов [140, 109].

В этом случае методика, позволяющая пользователю получать конкретное решение, основанное на практическом опыте предыдущих пользователей, решавших в прошлом аналогичную задачу (CBR), является альтернативой, восполняющей недостатки алгоритма, базирующего на интеграции FuzzyOWL и SWRL.

Таким образом, предложенная в данной работе методология построения нечетких онтологий с точки зрения организационного процесса в условиях сложной предметной области и наличия нескольких экспертов, описывающих её в терминах OWL, позволяет:

- существенно сократить временные издержки построения онтологии;
- избежать противоречий в интерпретации и формализации того или иного элемента предметной области;
- снизить субъективную составляющую оценки понятий ПрО отдельным экспертом, исходя из его компетентности в данной конкретной подобласти рассматриваемой области.

Все это возможно реализовать с помощью задания степени принадлежности всем определяемым объектам и отношениям, т.е. посредством внесения нечеткости в разрабатываемую онтологию.

Рассмотренная методология построения нечетких онтологий FuzzyOWL позволяет значительно расширить возможности использования онтологий в процессах получения новых знаний из уже определенных в ней, еще больше приблизить формализованное представление знаний о ПрО к естественному для

человека варианту их представления и восприятия.

Описанный в работе модифицированный алгоритм интеграции онтологической и продукционной форм хранения и представления знаний, благодаря введению нечеткости в элементы онтологии, снижает риски потери возможных результатов работы блока логического вывода основанной на данной базе знаний экспертной системе, а также увеличивает гибкость процесса вывода рекомендации, предоставляя пользователю более широкий выбор вариантов, упорядоченных по степени их релевантности.

4.5 Анализ эффективности семантической оценки терминологичности

Для оценки эффективности алгоритма автоматизированного обучения онтологии посредством извлечения знаний из текстов предметной области были проведены эксперименты по формированию базы знаний «Юнитех» одноименного учебного центра.

Учебный центр «Юнитех» — это платформа дистанционного взаимодействия студентов и преподавателей переводческих факультетов вузов, переводчиков, инженеров и работодателей. В настоящее время базой знаний данного переводческого учебного центра пользуется 20 ВУЗов по всей России.

База знаний «Юнитех» — это база элементарных знаний о наиболее распространенных технических объектах (механизмах, устройствах, системах и т.п.), которые являются общими для разных отраслей промышленности. Описание каждого объекта включает определение, назначение, конструкцию, принцип действия, правила эксплуатации и технического обслуживания, примеры взаимодействия с другими объектами.

В базе знаний объединены и описаны объекты, элементарные знания о которых имеются у опытных переводчиков и отсутствуют у начинающих. Цель построения базы знаний заключается в предоставлении студентам переводческих факультетов возможности изучения элементарных свойств наиболее распространенных технических объектов в таком минимальном объеме, который

будет давать максимально полезный эффект в условиях отраслевой неопределенности.

Ключевые особенности:

- Направленность на студентов переводческих факультетов вузов и начинающих переводчиков
- Отраслевые знания, оказывающие большое влияние на понимание технических текстов.

Таким образом, формируемая в результате проведения экспериментов по проверке эффективности разработанного алгоритма расширения онтологии за счет извлечения терминологии база знаний используется в учебных целях для дальнейшего точного перевода с учетом области применения терминологии.

4.5.1 Онтология «Станки с ЧПУ»

В рамках решения задачи анализа разработанных алгоритмов расширения ядра онтологии посредством извлечения терминологии из текста на основе семантической метрики «термин/ не термин» экспертами в выбранной области была разработана базовая OWL-онтология на основе руководства по эксплуатации токарно-фрезерного станка с числовым программным управлением (ЧПУ) [55].

Разработанная OWL-онтология имеет иерархическую организацию и включает в себя 318 классов, каждому из которых соответствует одноименный объект, и 17 свойств классов. На данный момент онтология имеет 4 уровня иерархии, что позволяет максимально конкретизировать термины предметной области, используемой при решении поставленной задачи. Фрагмент данной онтологии представлен на рисунке 4.3.

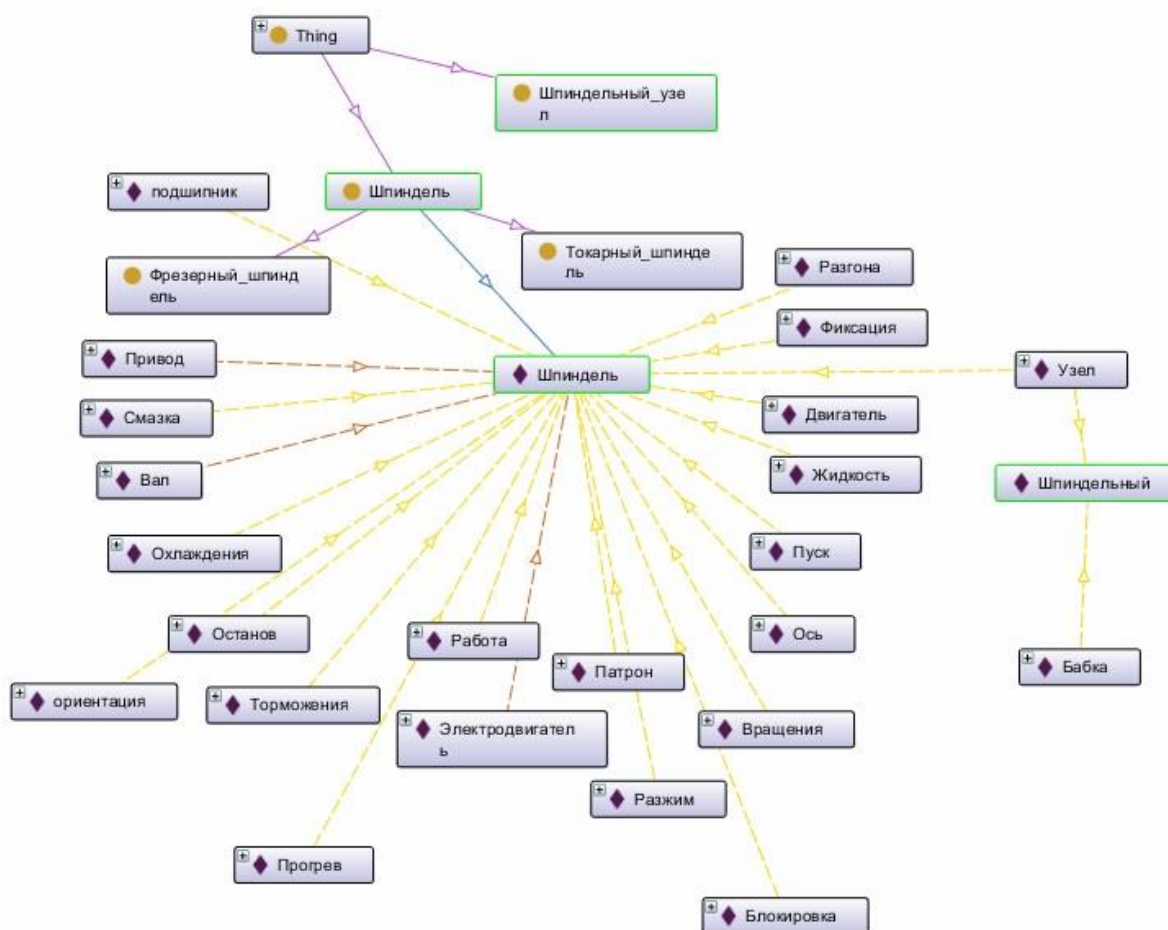


Рисунок 4.3 Фрагмент онтологии «Станки с ЧПУ»

Пример объявления класса «Концевая фреза» разработанной OWL-онтологии:

```
<owl:Class rdf:ID="КонцеваяФреза">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Концевая фреза</rdfs:label>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Фреза"/>
  </rdfs:subClassOf>
</owl:Class>
```

Для проведения эксперимента использовался текст объемом около 62000 слов из руководства по эксплуатации токарно-фрезерного станка с числовым программным управлением (ЧПУ) [46].

К особенностям текстов данной предметной области можно отнести высокую насыщенность терминами, влияние научного стиля на лексико-семантические,

морфологические, синтаксические параметры и формализованность содержания, опирающегося на логико-понятийную схему предметной области. Для оценки эффективности подсчёта показателей рассмотрены меры Precision, Recall и F_1 -мера для каждого показателя в каждой категории словоупотреблений.

В рамках проведения эксперимента были осуществлены следующие действия:

1. На основании специализированного справочника для станков с ЧПУ была разработана OWL-онтология в соответствии с моделью, описанной в Главе 2.
2. Для извлечения терминологии из текстовых массивов с целью формирования и дополнения базы знаний «Юнитех», использовался текст объемом около 62000 слов соответствующей предметной области из руководства по эксплуатации токарно-фрезерного станка с числовым программным управлением (ЧПУ).
3. На выходе разработанной системы онтологически-ориентированного извлечения терминологии, реализующей предложенный алгоритм, описанный в Главе 2, получился список n-словий.
4. Полученные сочетания оценивались экспертом на принадлежность к классу терминов рассматриваемой предметной области.
5. По итогам оценки высчитывалась эффективность предложенного алгоритма в соответствии с описанными ниже метриками [90].

4.5.2 Метрики оценки результатов

Рассмотрим метрики оценки степени терминологичности словосочетаний, применимые к задаче классификации вида «термин/не термин». Правильность попадания того или иного словосочетания в соответствующий класс оценивается экспертом в этой ПрО. Исходя из этого, ответы, полученные на выходе алгоритма, относящего словоупотребление к тому или иному классу, применительно к условиям поставленной задачи, выглядят следующим образом:

- Истинно-положительные («true positives», tp) — синтаксические

единицы, которые, с точки зрения эксперта, относятся к данному классу, и алгоритм их отнес к нему.

- Ложно-положительные («false positives», fp) — единицы, ошибочно, с точки зрения эксперта, отнесенные к классу.
- Ложно-отрицательные («false negatives», fn) — сочетания, ошибочно не отнесенные к данному классу.
- Истинно-отрицательные («true negatives», tn) — синтаксические единицы, которые верно, с точки зрения эксперта, не были определены к данному классу.

В этом случае мера точности (P, precision) определяется так:

$$P = \frac{tp}{tp + fp}$$

Мера точности показывает, сколько полученных от классификатора положительных ответов являются верными. Чем выше значение Precision, тем меньше синтаксических единиц, ошибочно отнесенных к классу терминов. Для оценки соотношения правильных и неправильных ответов, возвращенных классификатором, вычисляется мера полноты (R, recall):

$$R = \frac{tp}{tp + fn}$$

Мера полноты характеризует способность классификатора извлекать положительные ответы из ожидаемых.

Также для оценки эффективности классификатора используется унифицированная метрика F_1 :

$$F_1 = 2 \times \frac{P \times R}{P + R}$$

Фактически это среднее гармоническое величин P и R . Величина F_1 является одной из самых распространенных метрик для подобного рода систем. Именно F_1 используется, чтобы сформулировать пороговое качество разработанного семантического анализатора [3, 4].

4.5.3 Результаты вычислительных экспериментов извлечения терминов из проектной документации

Приводимые результаты экспериментов имеют целью изучение эффективности разработанных показателей. Всего будет рассмотрен результат работы двух показателей: тезаурусного и критерий вложенных связей; четырех категорий словоупотреблений: одиночных слов, двух-, трёх-, четырёхсловных словосочетаний.

Результаты экспериментов по извлечению терминов посредством применения тезаурусного критерия представлены в таблице 4.4, критерия вложенных связей – в таблице 4.5.

Так как в случае применения тезаурусного критерия оценивается терминологичность каждого слова/сочетания, поступающего на вход алгоритма, то для формального отделения терминов от не терминов в результате его выполнения, необходимо ввести пороговое значение $K_{Ont} = 0,5$.

Таблица 4.4 Результаты извлечения терминов с использованием тезаурусного алгоритма

Категория	Выделено экспертом в категории	Кол-во словоупотреблений $k_{Ont} > 0,5$	Из них терминов	Precision	Recall	F-мера
1	294	120	88	0,73	0,29	0,42
2	631	305	133	0,43	0,21	0,28
3	361	379	214	0,56	0,59	0,57
4	107	196	120	0,61	1,12	0,79

Таблица 4.5 Результаты извлечения терминов с использованием алгоритма вложенных связей

Категория	Выделено экспертом в категории	Кол-во терминов-кандидатов	Из них терминов	Precision	Recall	F ₁ -мера
1	294	168	154	0,91	0,52	0,66
2	631	431	372	0,86	0,58	0,69
3	361	370	327	0,88	0,9	0,89
4	107	159	129	0,81	1,2	0,97

Анализ результатов выполнения разработанных методик необходимо рассматривать с учетом различий вариантов словоупотреблений:

- **Одиночные слова.** Сравнение значений полученных характеристик представлено на рисунке 4.4.

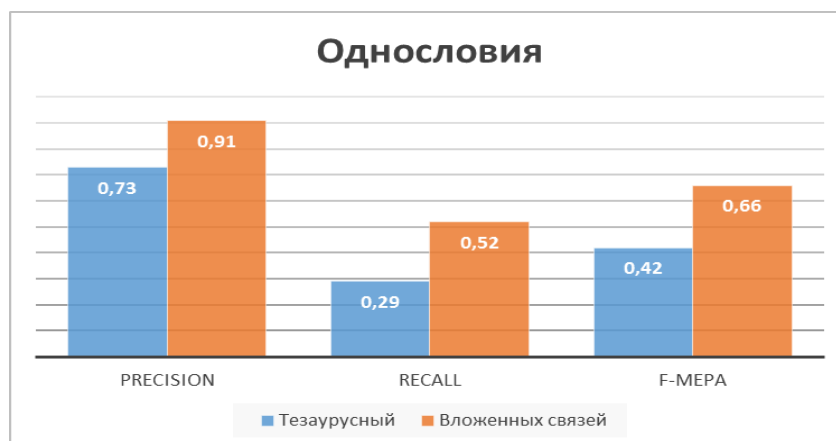


Рисунок 4.4 Сравнительная характеристика применения онтологических алгоритмов при извлечении однословий

Исходя из полученных выше результатов, следует отметить, что наилучшие показатели извлечения однословных терминов были получены при применении второго алгоритма. Почти все извлеченные алгоритмом одиночные слова являются терминами, в то время как всего было извлечено немногим более половины всех однословных терминов рассматриваемой проблемной области. Recall у тезаурусного показателя для однословных терминов хоть и ненамного ниже, но значение Recall позволяет судить о более низкой эффективности этого показателя. Таким образом, алгоритм вложенных связей оказался наиболее эффективным при вычислении однословных терминов, о чём свидетельствует и наивысшее значение F-меры среди показателей.

- **Двухсловные словосочетания.** Сравнение значений полученных характеристик представлено на рисунке 4.5.

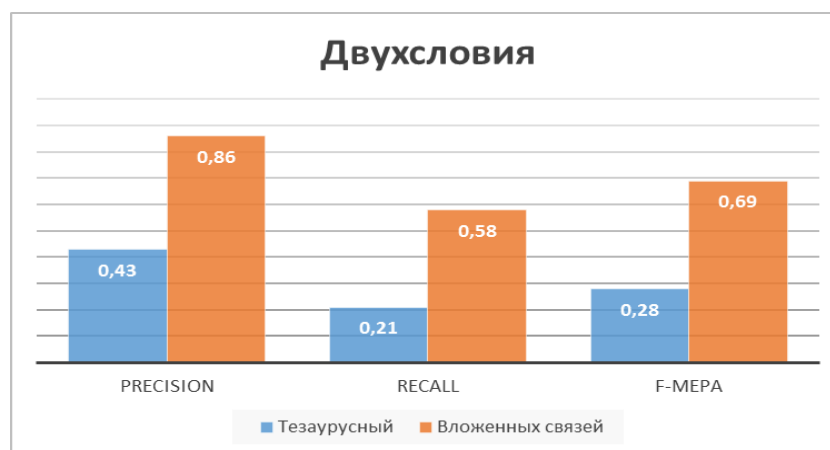


Рисунок 4.5 Сравнительная характеристика применения онтологических алгоритмов при извлечении двухсловий

Исходя из результатов анализа, можно сделать вывод, что тезаурусный алгоритм значительно уступает по полноте и точности второму критерию, имеющего лучшие значения Precision, Recall и F-меры среди всех результатов. При достаточно высокой точности критерий вложенных связей извлекает более половины двухсловных терминов предметной области. Таким образом, для извлечения двухсловных терминов наиболее эффективным также является второй алгоритм.

- Трехсловные словосочетания. Сравнение значений полученных характеристик представлено на рисунке 4.6.

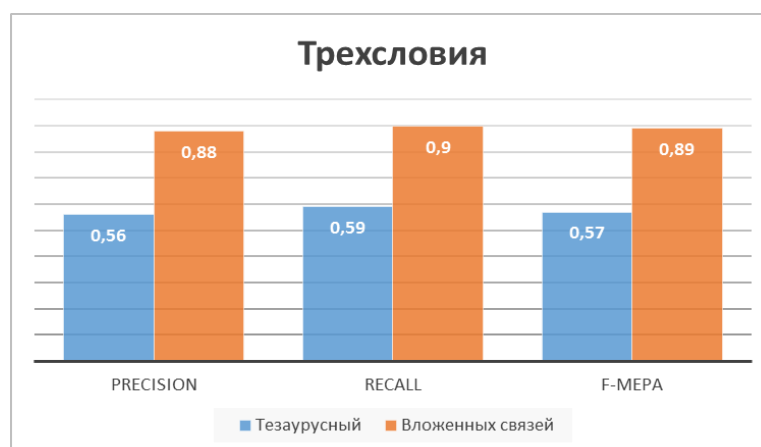


Рисунок 4.6 Сравнительная характеристика применения онтологических алгоритмов при извлечении трехсловий

Удовлетворительным можно считать результат работы тезаурусного алгоритма по извлечению трёхсловных терминов: извлекается более половины

трёхсловных терминов предметной области при среднем Precision. Результаты работы второго алгоритма можно назвать лучшими, о чём позволяют судить достаточно высокие значения Precision и Recall.

- Четырёхсловные словосочетания. Сравнение значений полученных характеристик представлено на рисунке 4.7.

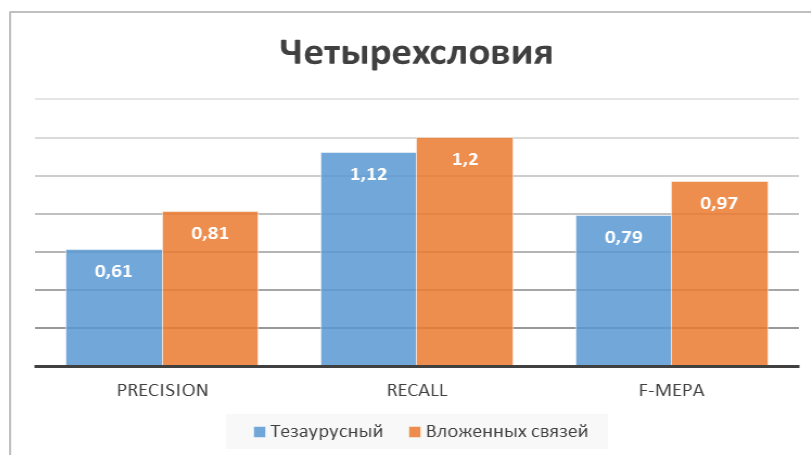


Рисунок 4.7 Сравнительная характеристика применения онтологических алгоритмов при извлечении четырехсловий

Тезаурусный алгоритм и алгоритм критерий вложенных связей оказались сопоставимыми по эффективности. Значение Recall больше 1 для обоих показателей говорит об извлечении ими терминов, ранее не выделенных в ходе экспертного анализа. Несмотря на схожие результаты, тезаурусный алгоритм все же проигрывает второму за счёт более низкого значения Precision. Таким образом, алгоритм вложенных связей оказался наиболее эффективным показателем для извлечения и четырехсловных терминов.

Полученные результаты экспериментов по извлечению терминов из инструкции по эксплуатации токарно-фрезерного станка с ЧПУ с использованием разработанной онтологии соответствующей предметной области позволяют сделать вывод о высокой эффективности использования алгоритма вложенных связей для решения поставленной задачи, особенно в случаях анализа трех и четырехсловий.

С целью проверки целесообразности применения каждого из предложенного онтологического алгоритма были проведены эксперименты по оценке точности

извлечения терминологии из текстов по предметной области «Станки с ЧПУ». При проведении экспериментов была использована специально составленная онтология предметной области и применена описываемая в данной главе метрика оценки результатов.

Для проведения эксперимента использовался текст объемом около 62000 слов из руководства по эксплуатации токарно-фрезерного станка с числовым программным управлением (ЧПУ), а также OWL-соответствующей предметной области, включающую в себя 261 экземпляр классов и порядка 746 отношений объектов классов.

Для оценки эффективности подсчета показателей рассмотрены меры Precision (P), Recall (R) и F_1 -мера в каждой категории словоупотреблений, а также проведены аналогичные эксперименты по извлечению терминов с использованием наиболее часто применяемых статистических методов: Frequency, TF*IDF, C-Value. Результаты экспериментов приведены в таблице 4.7.

Таблица 4.7 Извлечение терминологии с помощью статистических методов

Кол-во слов в словоупотреблении	Термины	Кандидаты	Верно	P	R	F_1
Онтологический алгоритм						
1	294	168	134	0,80	0,46	0,58
2	631	431	372	0,86	0,59	0,70
3	361	370	327	0,88	0,91	0,89
Frequency						
1	294	134	123	0,92	0,42	0,58
2	631	469	347	0,74	0,55	0,63
3	361	334	267	0,80	0,74	0,77
TF*IDF						
1	294	147	138	0,94	0,47	0,63
2	631	456	328	0,72	0,52	0,60
3	361	277	166	0,60	0,46	0,52
C-Value						
1	294	120	112	0,93	0,38	0,54
2	631	789	316	0,40	0,50	0,44
3	361	295	162	0,55	0,45	0,50

Исходя из полученных выше результатов, следует отметить, что статистические методы показали значительно более высокие результаты при извлечении однословий. Это связано с тем, что онтологический метод в первую очередь извлекает связанные с имеющимися в базе знаний термины. На

диаграммах 4.8 – 4.10 представлено сравнение значений характеристик Precision, Recall и F-меры алгоритма вложенных связей, а также статистических алгоритмов.

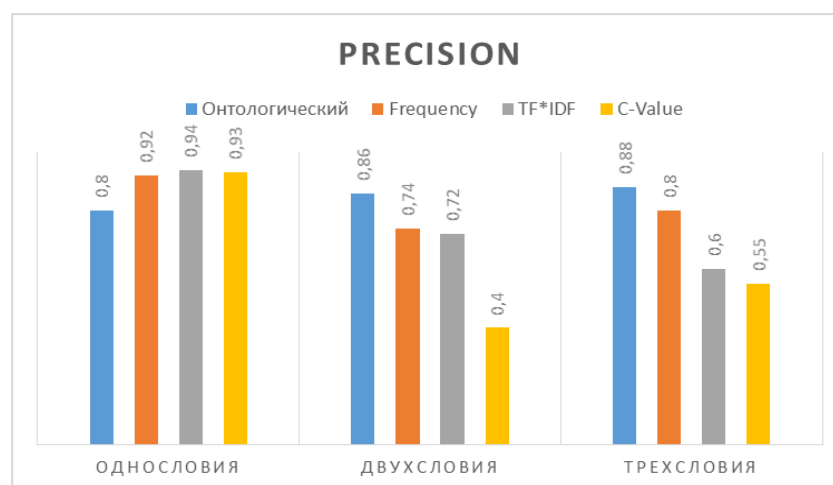


Рисунок 4.8 Значения показателя Precision

В случае же двух- и трехсловий, онтологический метод позволяет извлекать меньше неверных терминов. В первую очередь, это связано с тем, что статистические методы больше ориентируются на частоту употреблений словосочетаний, вне зависимости от привязки к особенностям ПрО, и могут извлекать общенаучные термины, а также термины других ПрО.

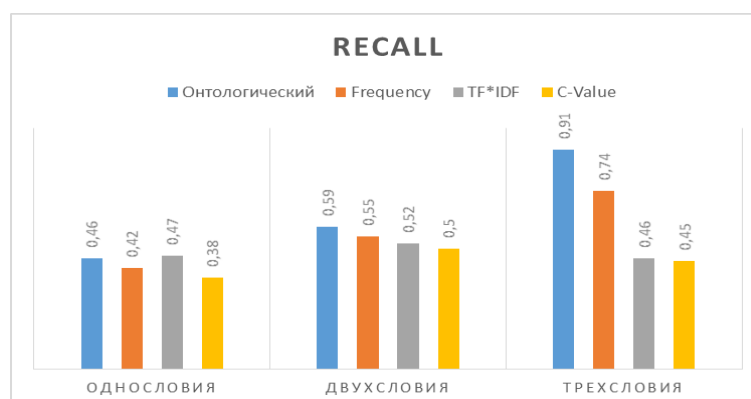


Рисунок 4.9 Значения показателя Recall

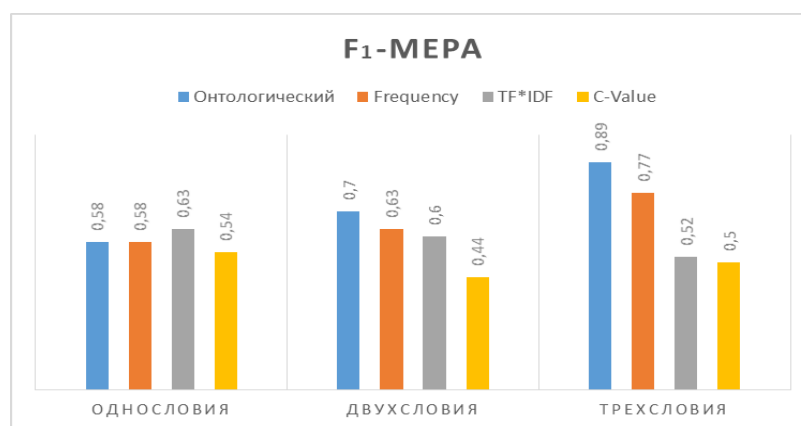


Рисунок 4.10 Значения показателя F-меры

4.5.4 Использование результатов работы предложенных алгоритмов при проведении отраслевой подготовки специалистов по мультязычной проектной документации

Процесс разработки проекта СТС предполагает наличие предварительно четко сформулированных требований к проектируемой системе. Унификация и формализация требований возможна лишь в случае проведения детального анализа предметной области применения проектируемой системы. Информационные ресурсы, которые позволяют проводить подобный детальный анализ, могут иметь различные формы представления: инструкции, интернет-ресурсы, учебно-методические материалы, предметные словари и тезаурусы и т.д. При этом данные ресурсы могут быть представлены на различных языках, вследствие чего появляется необходимость профессионального перевода ресурсов специалистами, разбирающимися в предметной области. Отсюда возникает потребность в экспресс-обучении специалистов подобного профиля с использованием специализированных учебно-методических материалов, охватывающих наиболее важные аспекты Про, необходимые в процессе перевода иностранных источников. Отсюда встает необходимость автоматизации формирования подобных методических ресурсов на основании имеющегося семантического ядра, отражающего основные принципы и законы проектной среды.

С целью оценки эффективности разработки единой базы терминологии, извлеченной из крупных корпусов текстов на основе предложенных

семантических алгоритмов, был проведен ряд экспериментов по оценке качества перевода проектной документации по тематике «Станки с ЧПУ».

Отраслевая подготовка специалистов подобного профиля – это комплекс обучающих мероприятий, направленных на формирование у специалиста системы отраслевых знаний в конкретной технической отрасли. Под системой отраслевых знаний понимается логически обоснованная система знаний в определенной предметной области, состоящая из множества понятий, их определений и аксиом, необходимых для ограничения интерпретации.

Эксперименты проводились на базе ООО Агентство переводов «Юнитранс». В течение 6 месяцев специалисты изучали «Учебный справочник технического переводчика: станки с ЧПУ», эксплуатационную документацию, видеоматериалы и дополнительную информацию в сети Интернет, выполняли рабочие переводы. Одним из пунктов отраслевой подготовки было изучение базы знаний «Юнитех», формируемой на основе применения предложенных семантических алгоритмов. В начале стажировки и по прошествии каждого месяца обучения стажеры выполняли перевод контрольных образцов, что позволило в динамике отследить показатели качества перевода.

Для описания исследования были выделены следующие аспекты:

Объект: специалисты-стажеры с высшим лингвистическим (филологическим) образованием без опыта работы.

Предмет: повышение качества технического перевода на примере предметной области «Станки с ЧПУ».

Цель: доказательство эффективности отраслевой подготовки.

Задачи:

1. Определить TQI испытуемых на разных этапах отраслевой подготовки.
2. Определить справочные значения.
3. Определить критические точки и сравнить результаты.
4. Выявить закономерности в характере повышения качества перевода на разных сроках отраслевой подготовки.
5. Доказать эффективность используемой методики отраслевой подготовки.

Методы исследования: качественный и количественный анализ по системе оценки TQI (Translation Quality Index).

Экспериментальная база:

1. 5 специалистов по работе с мультязычной документацией

В эксперименте участвовало 5 специалистов в возрасте 22-24 года с высшим образованием без опыта работы с мультязычной документацией.

2. 1 научно-технический редактор

Составление, проверка и оценка перевода контрольных образцов осуществлялась специалистом с высшим лингвистическим образованием и опытом работы более 3 лет, соавтором учебного справочника технического переводчика: станки с ЧПУ.

3. Справочные материалы

Учебный справочник был составлен сотрудниками компании после 3 лет работы по работе с англоязычной технической документацией к станкам с ЧПУ. Помимо справочника, использовалась специализированная база знаний «Юнитех», полученная с помощью применения алгоритма вложенных связей автоматизированного извлечения терминологии из специализированной литературы заданной тематики.

4. Методика отраслевой подготовки по тематике «Станки с ЧПУ»

Программа отраслевой подготовки рассчитана на 6 месяцев и состоит из 2 этапов.

1 этап (3 месяца):

- изучение справочных материалов: базы знаний «Юнитех», а также справочника по тематике «Станки с ЧПУ» – 1 тема в день: получение основных знаний по устройству и принципам работы металлообрабатывающего оборудования;
- просмотр видеоматериалов: наглядное представление о выполнении различных видов обработки на станках токарно-фрезерной группы;
- изучение эксплуатационной документации к станкам с ЧПУ на русском языке: развитие навыка «понимание-вспоминание».

2 этап (3 месяца):

- повторение пунктов 1 этапа: построение отраслевых знаний в иерархическую систему;
- изучение учебников: углубление и расширение отраслевых знаний, развитие навыка «понимание-вспоминание»;
- изучение актуальных статей: углубление и расширение отраслевых знаний, развитие навыка «понимание-вспоминание» + получение дополнительной информации в сети Интернет.

5. Количественная оценка качества перевода (TQI)

Для оценки качества перевода технической документации использовалась система TQI, в которой оценка производится по количеству допущенных ошибок на единицу перевода. В используемой методике расчета TQI идеальным считается перевод с индексом 0.

Разработанная классификация ошибок, положенная в основу TQI, охватывает типы ошибок, влияющие на качество инженерно-технического перевода. С этой точки зрения интерес представляют ошибки отраслевых знаний (терминологические, смысловые, перенос ошибки из оригинала в перевод, ошибки перевода устойчивых словосочетаний).

При расчете TQI учитывается размер текста, вес ошибки, а также количество потенциальных ошибок каждого типа.

6. Контрольный текст

Основой эксперимента послужили 7 контрольных образцов технического текста по предметной области «Станки с ЧПУ» на английском языке. Они представляют собой выдержки из руководств по эксплуатации, техническому обслуживанию и программированию, оформленные в виде законченного текста объемом 18000 знаков с пробелами. В них отражены основные технические аспекты, которые встречаются в эксплуатационной документации.

Специалисты работали с технической документацией на английском языке в начале обучения (КТ № 0) и после каждого месяца обучения (КТ №№ 1-6).

Все контрольные тексты имеют примерно одинаковую сложность и

характеризуются относительно одинаковой терминологической насыщенностью (от 170 до 200 терминов, что корректируется соответствующим коэффициентом при расчете TQI). Кроме того, в контрольных образцах были намеренно оставлены (либо искусственно созданы) ошибки отраслевых знаний (2 грубых смысловых ошибки и 3 негрубых).

7. Справочные значения

Справочные значения являются опорными данными, в сравнении с которыми будут интерпретироваться результаты эксперимента.

Справочным значением № 1 является индекс качества перевода, выполненного специалистом с опытом работы более 2 лет в другой отрасли. Для получения данного справочного значения было задействовано два лучших специалиста по работе с технической документацией к нефтегазовому оборудованию для перевода одного контрольного образца. В результате средний показатель составил TQI 155.

Справочным значением № 2 является индекс качества перевода того же контрольного образца, выполненного теми же специалистами, но с использованием согласованной терминологии. Объем согласованной терминологии составляет 74 % от общего количества терминов. В результате был получен средний TQI 71.

Справочным значением № 3 является индекс перевода приемлемого качества. Перевод приемлемого качества характеризуется незначительными (редкими и легко преодолимыми) сложностями понимания смысла целевым специалистом (буквальный перевод отдельных единиц, описывающих редкое оборудование, негрубые терминологические и смысловые ошибки, не искажающие смысл). TQI перевода приемлемого качества не должен превышать 30. При превышении этого значения у целевого специалиста может складываться недоверие к его компетентности (чуждая терминология, конструкция устойчивых выражений, нарушение логики, абсурдные смысловые ошибки и т.п.).

Справочным значением № 4 является индекс перевода высокого качества. Он был получен в результате проверки 3 переводов, выполненных специалистами с

опытом работы более 3 лет по тематике «Станки с ЧПУ». Его среднее значение составляет 10. В переводе высокого качества допускается совершение на 18000 знаков перевода не более 5 ошибок отраслевых знаний с незначительным влиянием на технический смысл. Перевод высокого качества обеспечивает быстрое и легкое понимание смысла текста целевым специалистом.

Результаты отраслевой подготовки в сравнении со справочными значениями по итогам проведенных экспериментов приведены на рисунке 4.11.

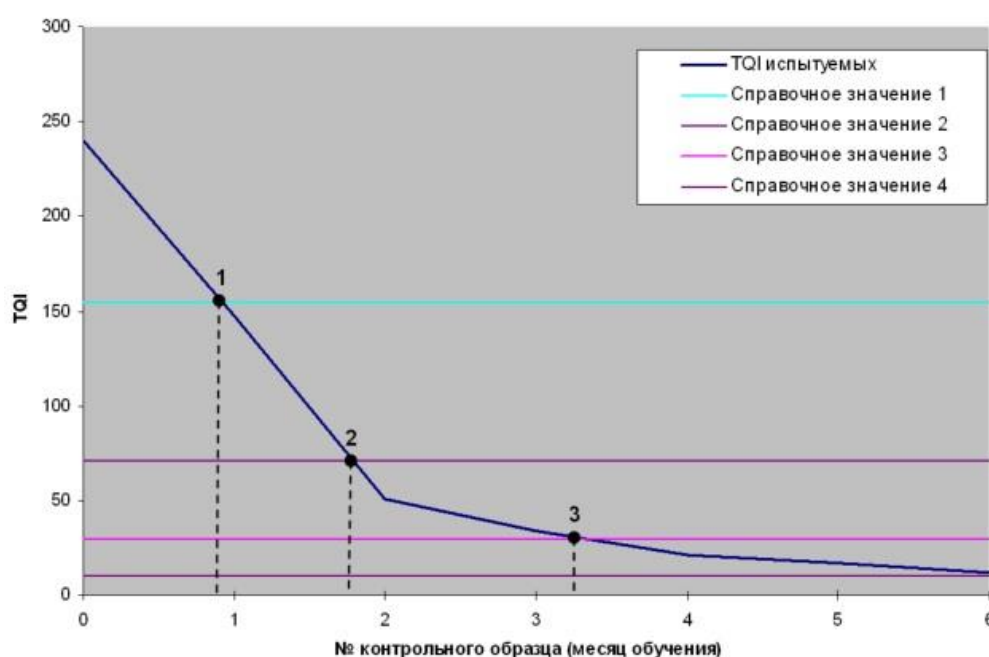


Рисунок 4.11 Результаты отраслевой подготовки в сравнении со справочными значениями

По вертикальной оси данного графика представлены значения TQI, по горизонтальной – номера контрольных текстов (КТ 0 – в начале обучения, КТ 1-6 по окончании соответствующего месяца обучения).

Данный график предназначен для наглядного представления результатов обучения в зависимости от его срока.

Кривая синего цвета обозначает TQI испытуемых, исходным значением которого является индекс 240 (средний результат пяти переводов КТ № 0).

1) критическая точка 1 – пересечение линии TQI переводчика с линией справочного значения 1 (индекс качества перевода, выполненного переводчиком с опытом работы более 2 лет в другой отрасли). При проекции этой точки на ось X видно, что данный результат достигается в течение 1 месяца отраслевой

подготовки. В итоге, будучи в равных условиях (без использования согласованной терминологии), стажер с опытом работы менее 1 месяца и переводчик с опытом работы более 2 лет показывают одинаковый результат.

2) критическая точка 2 – пересечение линии TQI специалиста с линией справочного значения 2 (индекс перевода того же контрольного образца, выполненного теми же опытными специалистами, но с использованием согласованной терминологии).

При проекции этой точки на ось X видно, что для достижения этого результата стажеру достаточно менее 2 месяцев обучения. Получается, что, даже не пользуясь базой согласованных терминов, начинающий специалист уже за 2 месяца обучения не будет уступать более опытному специалисту, который использовал согласованную терминологию.

3) критическая точка 3 – пересечение линии TQI с линией справочного значения 3 (индекс перевода приемлемого качества). При проекции этой точки на ось X видно, что данный результат достигается в течение 3-4 месяцев отраслевой подготовки. Таким образом, можно очень быстро перейти к полноценной профессиональной деятельности, будучи вооруженным основами изучаемой отрасли. За такой срок обучения можно освоить всю необходимую теоретическую базу, а затем быстро прогрессировать, опираясь на полученные знания.

4) критическая точка 4 – пересечение линии TQI с линией справочного значения 4 (индекс перевода высокого качества). Как видно по графику, уже через полгода обучения специалист может показывать практически безупречный результат, такой же, как и у опытного специалиста.

Выводы:

1. Качество перевода, выполненного специалистом с опытом работы более 2 лет, но в другой предметной области, в 5 раз ниже приемлемого и в 15 раз ниже высокого.

2. Использование согласованной терминологии не спасает ситуацию – качество все равно в 2,36 раза ниже приемлемого, и в 7 раз ниже высокого.

3. Получение отраслевых знаний оправдано тем, что позволяет специалисту

добиться приемлемого качества перевода за 3 месяца, а высокого – за 6-8.

Результаты зависимости количества приобретенных отраслевых знаний от месяца подготовки представлены на рисунке 4.12.

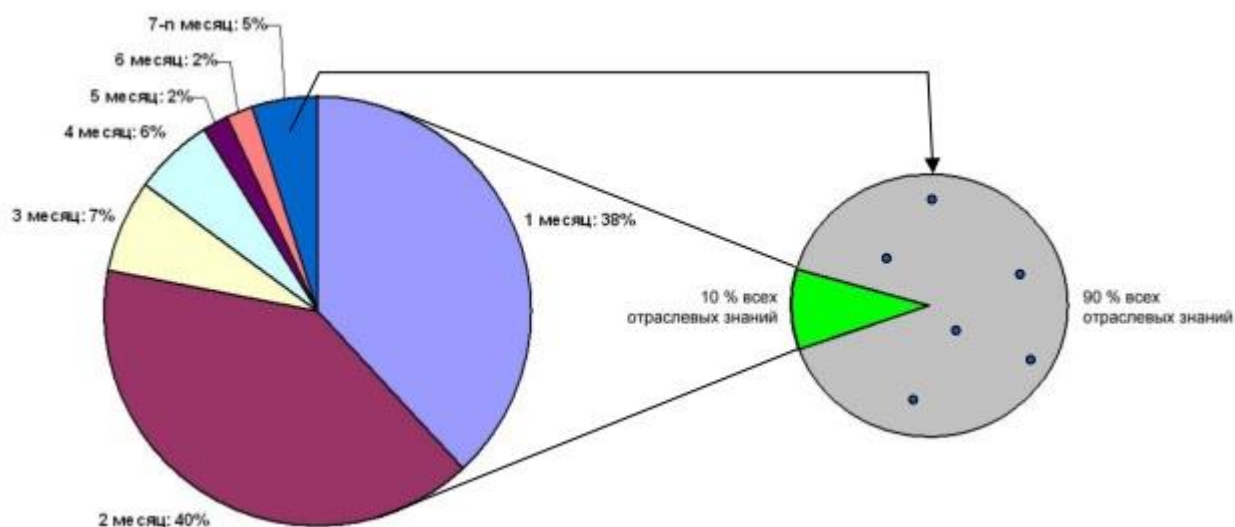


Рисунок 4.12 Зависимость количества приобретенных отраслевых знаний от месяца подготовки

При расчете данных значений было принято утверждение, что до начала подготовки количество отраслевых знаний, полученных по программе, составляет 0 %, и выражается в TQI 240, а количество максимальных отраслевых знаний составляет 100 % и выражается в TQI 0.

Как видно, основная доля приобретенных отраслевых знаний приходится на первые 2 месяца (78 %). Это обусловлено окончанием изучения базы знаний «Юнитех», а также учебного справочника технического переводчика: станки с ЧПУ, которое приходится на конец второго месяца.

Выводы:

1) Подавляющий объем отраслевых знаний специалист получает из полученной с помощью применения разработанных алгоритмов универсальной базы знаний «Юнитех», а также вспомогательного справочника (85 % - результат 3-го месяца подготовки, когда отраслевые знания образуют систему, и 91 % – результат 4-го месяца подготовки, когда специалисты заканчивают повторение учебного справочника).

2) Поскольку прогресс не останавливается после 4-го месяца, можно сделать вывод о том, что отраслевые знания, полученные из базы знаний «Юнитех» и

справочника, служат основой для получения более глубоких отраслевых знаний из других источников.

3) Информация в базе знаний подается в форме, удобной для восприятия специалистом по работе с мультязычной документацией (процентное распределение за 1 и 2 месяца высокое и примерно одинаковое (38 % и 40 %)).

4) Как видно, после прохождения программы отраслевой подготовки специалистам не достает лишь 5 % необходимых отраслевых знаний, которые они будут получать очень долго и до конца не получают, т.к. эти требуемые 5 % знаний динамичны и являются какой-то частью знаний всей отрасли.

Необходимо помнить, что получаемые по программе отраслевой подготовки знания не являются исчерпывающими. К предметной области «станки с ЧПУ» относятся десятки дисциплин и сотни специализированных учебников. Таким образом, программа отраслевой подготовки позволяет сосредоточиться на получении исключительно полезных специалисту знаний, которые обеспечивают правильное понимание и передачу 95 % отраслевой информации, содержащейся в переводимой эксплуатационной документации к станкам с ЧПУ.

4.6 Применение АС поддержки проектирования в деятельности АО «Авиастар-СП»

Проверка эффективности предложенных алгоритмов представления терминологии была проведена посредством ее применения в рамках разработки рабочей документации по объекту «Реконструкция и техническое перевооружение для создания поточной линии сборки самолетов для производства тяжелого военно-транспортного самолета Ил-76МД- 90А на АО «Авиастар-СП» [106].

Автоматизированная система управления поточной линией сборки (АСУ ПЛС) создаётся как составная часть в составе интегрированной автоматизированной системы информационной поддержки жизненного цикла воздушного судна (ВС) гражданской и транспортной авиации на основе электронного определения изделия (ИАС) на основе действующих в АО

«Авиастар СП» подсистем конструкторско-технологической подготовки производства (КТПП) ИАС и развития подсистем Управления Комплектацией Сборочного Производства и Системы Технологического Моделирования Процессов «ТЕМП2».

В общем случае, АС предназначена для автоматизации деятельности структурных подразделений, реализующих бизнес-процессы в соответствии с Таблицей 4.8.

- Служба главного конструктора;
- Службы главного технолога и главных специалистов;
- Дирекция по производству и цехи сборочного производства;
- Цехи производства технологического оснащения;
- Подразделения коммерческих служб, занимающиеся закупками и размещением заказов на проектирование, изготовление средств технологического оснащения (СТО) и другими видами кооперации;
- Дирекция по качеству и сертификации.

Таблица 4.8 Автоматизированные процессы

Автоматизируемые бизнес-процессы	Структурные подразделения					
	1	2	3	4	5	6
Конструкторская подготовка производства	ДА	ДА	ДА	ДА	ДА	ДА
Централизованной технологической подготовки производства		ДА	ДА	ДА		ДА
Проектирования, изготовления и эксплуатации СТО		ДА	ДА	ДА	ДА	ДА
Технологической подготовки в цехах сборочного производства		ДА	ДА	ДА		ДА
Производственно-технологическое моделирование	ДА	ДА	ДА	ДА	ДА	ДА
Управление комплектацией сборочного производства ВС			ДА	ДА	ДА	ДА
Планово-предупредительное обеспечение рабочих мест	ДА	ДА	ДА	ДА	ДА	ДА
Мониторинга и анализа состояния конструкторского, технологического, производственного процессов	ДА	ДА	ДА	ДА	ДА	ДА
Взаимодействие с предприятиями-кооперантами	ДА	ДА	ДА	ДА	ДА	ДА

Целями работы системы являются:

- управление поточной линией сборки в автоматизированном режиме;
- повышение качества процессов сборки изд. ИЛ-76-МД ;
- обеспечение сокращения времени на подготовку процессов окончательной сборки изд. ИЛ-76-МД;
- повышение эффективности сборочных работ на рабочих станциях поточной линии по изд. ИЛ-76-МД.

4.6.1 Характеристики объекта автоматизации

Объектом автоматизации являются основные бизнес-процессы, связанные с функционированием рабочих станций поточной линии сборки изд. ИЛ-76-МД.

Отдельные рабочие станции, как объекты в составе линии сборки, могут иметь встроенные средства автоматизации в виде специальных управляющих контроллеров или систем ЧПУ. Описание встроенных средств автоматизации, систем программного управления рабочими станциями, программных средств и необходимых входных данных, необходимых для разработки управляющих программ и лицензии на необходимое количество ПО входят в состав документации конкретных рабочих станций.

Разрабатываемая АС в соответствии с описанием рабочих станций должна обеспечивать подготовку необходимых данных на вход управляющих контроллеров или систем ЧПУ автоматизированных рабочих станций, а также обеспечить приём и хранение соответствующих выходных данных.

Основными функциями АСУ ПЛС являются:

- планирование обеспечения ресурсами рабочих станций;
- контроль за расходом ресурсов;
- графическая индикация и контроль подачи комплектующих для сборки на рабочую станцию;
- генерация необходимой отчетности.

Процесс сборки и испытаний самолета Ил-76МД-90А на предприятии осуществляется по следующей технологической схеме, в указанном порядке по станциям (таблица 4.9).

Таблица 4.9. Перечень станций и краткое определение работ на них

Обозначение станции	Наименование станции	Описание основных операций, выполняемых на станции
ПС.10	Стыковка фюзеляжа	Стыковка секций Ф1, Ф2, Ф3 в фюзеляж, монтаж шасси, проверка герметичности и опрессовка фюзеляжа
ПС.20	Станция монтажа гидросистем	Монтаж трубопроводов гидросистемы
ПС.30	Станция электромонтажей	Монтаж электрожгутов
ПС.40	Станция стыковки крыла	Стыковка крыла, испытание герметичности бака, монтаж десантного, ПНК, бытового оборудования
ПС.50	Станция комплектации шасси	Комплектация и подготовка к установке шасси
ПС.60	Станция стыковки киля со стабилизатором	Разделка киля, стыковка киля со стабилизатором
ПС.70	Станция подготовки двигателя	Подготовка двигателей к навеске на самолет
ОС.10	Станция навески двигателей	Стыковка оперения, навеска двигателей, монтаж топливной системы, системы управления, ПОС, СКВ, ППЗ, НГ, электрооборудование в крыле
ОС.20	Станция монтажа и отработки систем	Отработка системы электроснабжения, промывка и отработка гидросистемы, отработка высотного оборудования, кислородной системы, системы управления самолетом, ППЗ, НГ, окончательный монтаж ТЗИ
ОС.30	Станция финальной отработки	Испытания десантного, транспортного, аварийно-спасательного оборудования, комплектация изделия, отработка на КИС, предъявления, сдача на ЛИС
ЛГ.10	Логистическая станция	Логистика и другое вспомогательное оборудование для прочих станций
АП.10	Мероприятия по агрегатно-сборочному производству	В эту условную станцию включены мероприятия по модернизации оснастки АСП для снижения технических рисков по стыковке КЧК

Каждая рабочая станция, являющаяся звеном в единой цепи сборки агрегатов, на вход получает массив данных, на основании которых специалист принимает решение о возможности запуска процесса сборки на конкретной рабочей станции. Схематично данный процесс представлен на рис.4.13.

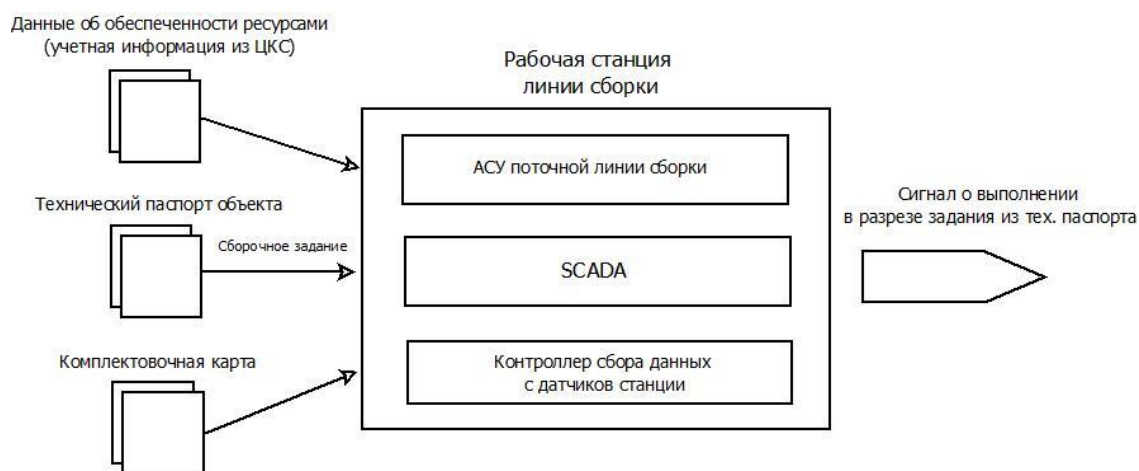


Рисунок 4.13 Схема потоков данных на АСУ ПЛС конкретной рабочей станции

Как видно из данной схемы, на вход АСУ ПЛС рабочей станции помимо комплектующих и агрегата, подается информация об обеспеченности необходимыми ресурсами в соответствии с комплектовочной картой, а также непосредственное сборочное задание.

На самом нижнем уровне системы управления технологическим процессом на каждой конкретной станции находится программируемый контроллер сбора данных с датчиков станции в виде сигналов. Далее полученная контроллером информация обрабатывается системой SCADA (Supervisory Control And Data Acquisition). Основные функции SCADA:

- Логическое управление.
- Обмен данными с промышленными контроллерами и платами ввода-вывода в онлайн-режиме;
- Визуальное представление полученной и обрабатываемой информации;
- Работа с БД, содержащей технологическую информацию.
- Аварийная сигнализация и управление тревожными сообщениями.
- Формирование отчетов относительно хода ТП.
- Обеспечение взаимосвязи с внешними программными системами (электронные таблицы, текстовые процессоры, СУБД и т. д.).

Верхний уровень системы занимает АСУ ПЛС, целью которой является окончательная обработка и формализация поступивших с нижних уровней данных, управление конкретными операциями и процессами, выполняемыми данной станцией, а также обеспечение человеко-машинного интерфейса (HMI).

Система должна иметь структуру, представленную на рисунке 4.14 или аналогичную ей по функциональным возможностям.

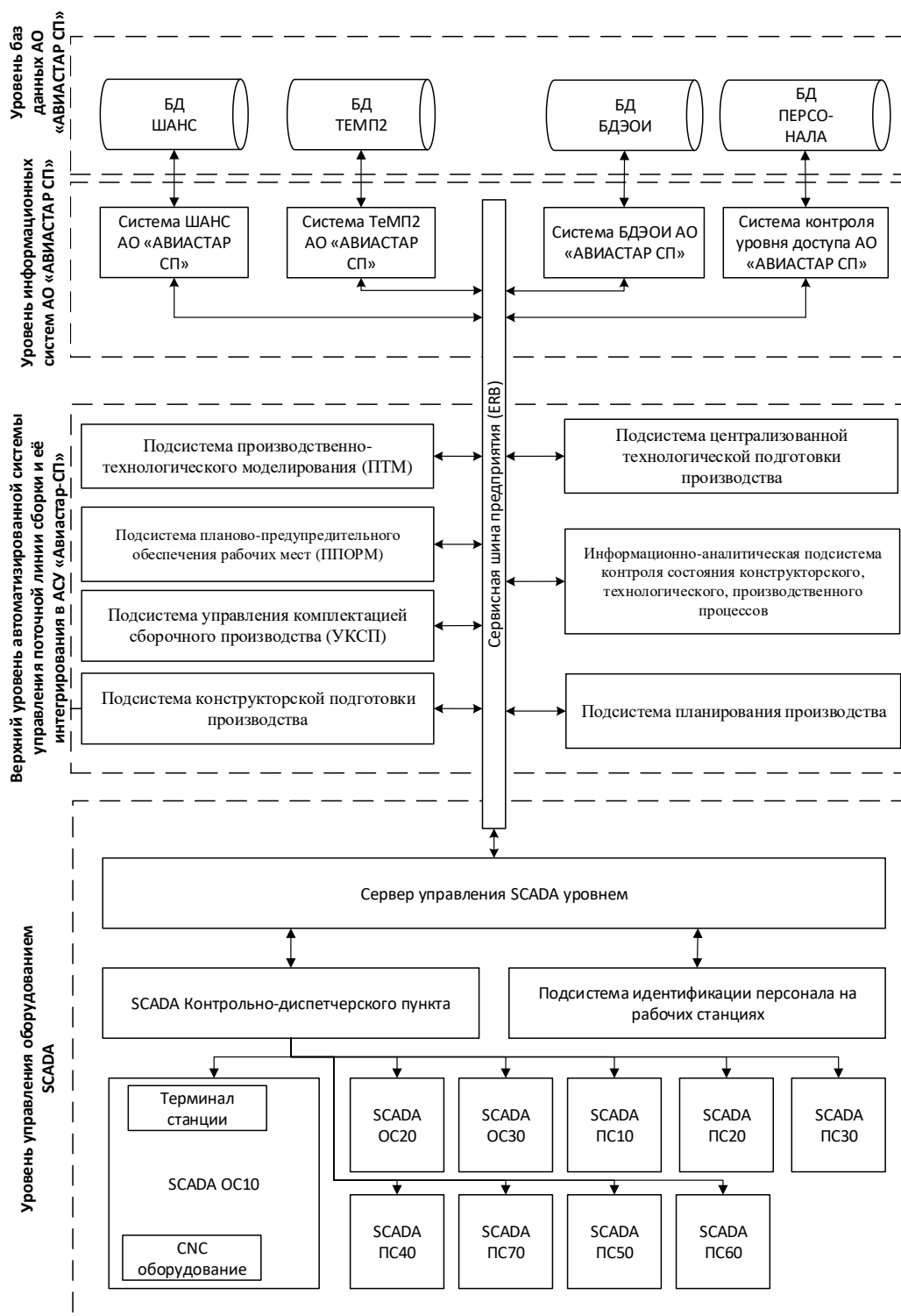


Рисунок 4.14 Общая структура АСУ ПЛС

4.6.2 Направления применения разработанной системы

При проектировании сложных технических систем таких, как АСУ ПЛС, по мере накопления опыта решения задач на отдельных этапах проектирования, появляется необходимость формирования отдельной составляющей информационного обеспечения САПР, определяющей терминологическое

окружение всего технологического процесса. В связи с тем, что разработка унифицированной документации по АСУ ПЛС предполагает наличие глубоких знаний в области самолетостроения, разработка единого формализованного информационного ресурса в виде онтологических структур по процессу сборки самолета обеспечивает удовлетворение одного из важных требований при использовании САПР в практике проектирования – наличие полной информации о проектируемом объекте. Разными специалистами, в зависимости от квалификации и технического кругозора, названные выше процедуры формулируются по-разному, а интерпретации одного и того же термина напрямую зависят от сложившегося за многие годы понятия.

Использование прикладной онтологии предполагает решение следующих задач:

- обеспечение общего терминологического пространства для совместного использования и понимания всеми пользователями;
- определение точного и непротиворечивого значения каждого термина;
- задание семантики ТП посредством определения множества аксиом, позволяющих генерировать новые знания относительно особенностей ПрО из уже имеющихся.

Онтология определяет язык, при помощи которого выражается семантическая часть словаря.

С целью унификации терминологии в процессе разработки технической документации АСУ ПЛС АО «Авиастар-СП» была разработана предметная онтология, состоящая (на данный момент) из 140 классов и их объектов, а также 26 свойств типа данных и свойств объектов.

Порядок проведения экспериментов:

1. Без помощи разработанной системы была составлена начальная версия ТЗ по соответствующему проекту.
2. Было разработано ядро OWL-онтологии по предметной области АСУ Поточной линии сборки тяжелых самолетов. Ядро состоит из 140 классов и объектов классов, а также 26 отношений и имеет 3 уровня иерархии.

3. Было выбраны наиболее часто используемых в процессе обучения специалистов в данной области ресурсы и учебные пособия, среди которых:

- Колганов И.М., Филиппов В.В. Проектирование приспособлений, прочностные расчеты, расчет точности сборки: Учебное пособие. - Ульяновск: УлГТУ, 2000. - 99 с.
- Ершов, В. И. Технология сборки самолётов: учебник для студентов авиационных специальностей вузов / В. И. Ершов, В. В. Павлов, М. Ф. Каширин, В. С. Хухорев. – М.: Машиностроение, 1986. – 456 с.

4. С целью исправления семантических ошибок и расширения терминологической составляющей ТЗ, к данным ресурсам были применены разработанные алгоритмы онтологического расширения ядра онтологии на основе извлечения терминов. Лучшие результаты показал алгоритм вложенных связей, эффект использования результатов применения которого в процессе разработки промежуточных и конечных версий ТЗ приведен в табл.3 и на рис. 4.15. В качестве характеристик, определяющих семантическую адекватность разработанного ТЗ были выделены следующие показатели: количество используемых в тексте ТЗ определений и число исправленных смысловых ошибок.

Таблица 4.10 – Результаты экспериментов по использованию тезаурусного алгоритма в процессе разработки ТЗ АСУ Поточной линии сборки

Характеристика	Начальная версия	v4	v9	v13	v16
Количество смысловых ошибок	13	3	2	3	-
Количество определений	9	15	16	18	20



Рисунок 4.15 График снижения количества смысловых ошибок в ЧТЗ АСУ ПЛС

Как видно из результатов экспериментов, применение разработанных алгоритмов уже после первой итерации позволило значительно снизить количество семантических ошибок в тексте ТЗ, а количество определений, включенных в структуру ТЗ, возросло в 1,7 раза.

В заключении сформулированы основные выводы и результаты диссертационной работы.

4.7 Выводы по главе

В четвертой главе представлены результаты экспериментов по применению предложенных алгоритмов, реализованных в интеллектуальной компоненте САПР, на базе процесса сборки самолетов для производства тяжелого военно-транспортного самолета Ил-76МД- 90А на АО «Авиастар-СП».

В рамках исследования были проведены эксперименты по анализу состояния ЛВС в условиях искусственного повышения трафика с использованием разработанной онтологически-ориентированной системы логического вывода, реализующей предложенные модели и алгоритмы интеграции и обучения экспертной базы знаний. Также проведен сравнительный анализ результатов экспериментов с использованием четких и нечетких онтологий. По итогам проведенных экспериментов можно сделать вывод о большей эффективности использования нечетких онтологий по сравнению с

четкими вследствие предоставления пользователю большего набора возможных решений, ранжированных по степени уверенности.

Помимо этого, представлены результаты экспериментов по извлечению терминов из проектной документации с помощью разработанных алгоритмов при построении универсальной базы знаний учебного центра «Юнитех» агентства переводов «Юнитранс» с использованием разработанной предметной онтологии «Станки с ЧПУ». Предложенный алгоритм вложенных связей показал лучшие результаты по извлечению многословных терминов предметной области в сравнении с статистическими методами.

Также представлены эксперименты по применению предложенных алгоритмов интеграции различных форматов представления экспертных знаний в процессе валидации терминологии в процессе разработки документации Автоматизированной системы управления Поточной линии сборки АО «Авиастар-СП».

Заключение

В ходе диссертационного исследования получены следующие результаты:

1. проведен сравнительный анализ современных средств представления терминологии в лингвистическом обеспечении интеллектуальных САПР;
2. разработана онтологически-ориентированная система извлечения терминологии из проектных документов с использованием сформированного ядра предметной онтологии;
3. проведен сравнительный анализ современных интеллектуальных методов логического вывода в САПР, выявить их возможности и ограничения в плане решения задач автоматизированного проектирования;
4. рассмотрена возможность применения методов онтологического анализа для решения задач оценки состояния сложного технического объекта рассматриваемой предметной области с целью поддержки принятия проектных решений на экспертном уровне;
5. исследована возможность интеграции методов теории нечетких систем и онтологического анализа состояния сложной технической системы;
6. разработана онтологическая модель представления набора продукций и алгоритм логического вывода рекомендаций на экспертном уровне в задачах автоматизированного проектирования;
7. разработан алгоритм расширения процедурной составляющей базы знаний интеллектуальной САПР с использованием механизма прецедентов;
8. разработан алгоритм расширения декларативного описания предметной области (ПрО) с использованием сформированного ядра онтологии интеллектуальной САПР посредством извлечения терминологии из проектных документов;
9. разработана АС поддержки проектирования, реализующую алгоритм интеграции механизмов логического вывода и онтологического представления экспертных знаний с элементами нечеткости, а также использующей механизм учета прецедентов с целью расширения ядра онтологии;

10. проведены вычислительные эксперименты, основными результатами которых являются:

- Применение разработанных алгоритмов интеграции продукционной и онтологической форм представления предметной терминологии с элементами нечеткости позволяет снизить риски потери возможных результатов работы блока логического вывода, а также увеличивает гибкость процесса вывода рекомендации, предоставляя пользователю более широкий выбор вариантов, упорядоченных по степени их релевантности (более, чем в 5 раз расширен список возможных решений)
- Использование в процессе обучения специалистов по работе с мультязычной проектной документацией унифицированной базы знаний, полученной путем извлечения терминологии предложенными онтологическими алгоритмами, позволяет обеспечить специалиста 78% приобретаемых отраслевых знаний за первые два месяца обучения.
- Использование алгоритмов извлечения терминологии из проектной документации по тематике станков с ЧПУ на основе предметной онтологии повышает эффективность извлечения двух- и трехсловных терминологических сочетаний на 20,5% и 33% соответственно.

11. результаты исследований внедрены в практику процесса проектирования технических систем предприятий региона.

Библиографический список

- 1 Aamodt, A. Case-based reasoning: foundational issues, methodological variations, and system approaches/ A. Aamodt, E. Plaza// Communications. – 1994. – № 7. – С. 39–59.
- 2 Ahmad K. et al. University of Surrey Participation in TREC8: Weirdness Indexing for Logical Document Extrapolation and Retrieval (WILDER) //TREC. – 1999.
- 3 Ananiadou S. A methodology for automatic term recognition //Proceedings of the 15th conference on Computational linguistics-Volume 2. – Association for Computational Linguistics, 1994. – С. 1034-1038.
- 4 Biber D., Conrad S., Reppen R. Corpus linguistics: Investigating language structure and use. – Cambridge University Press, 1998.
- 5 Bobillo F., Straccia U. Fuzzy description logics with general t-norms and datatypes // Fuzzy Sets and Systems 160 (23) – 2009 — p. 3382–3402.
- 6 Bobillo F., Straccia U. Fuzzy ontology representation using OWL 2 // International Journal of Approximate Reasoning. Vol. 52 – 2011. – p. 1073–1094.
- 7 Bourigault D. Surface grammatical analysis for the extraction of terminological noun phrases //Proceedings of the 14th conference on Computational linguistics-Volume 3. – Association for Computational Linguistics, 1992. – С. 977-981.
- 8 Cheng Gang, Du Qingyun. The Design and implementation of ontology and rules based knowledge base for transportation. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.Vol. XXXVII. Part B2. Beijing 2008.
- 9 Daille B. Approche mixte pour l'extraction de terminologie: statistique lexicale et filtres linguistiques : дис. – 1994.
- 10 David B. Leake (Ed.), Case-Based Reasoning: Experiences, Lessons, and Future Directions. Menlo Park , CA : AAAI Press/MIT Press, 1996, ISBN 0-262-62110-X.

- 11 Dunning T. Accurate methods for the statistics of surprise and coincidence //Computational linguistics. – 1993. – Т. 19. – №. 1. – С. 61-74.
- 12 Frantzi K., Ananiadou S., Mima H. Automatic recognition of multi-word terms: the c-value/nc-value method //International Journal on Digital Libraries. – 2000. – Т. 3. – №. 2. – С. 115-130.
- 13 Gruber T. R. A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220, 1993.
- 14 Isabel L. Nunes and M'ario Simões-Marques Applications of Fuzzy Logic in Risk Assessment The RA_X Case, Fuzzy Inference System Theory and Applications, Dr. Mohammad Fazle Azeem (Ed.), 2012. – pp. 21-40.
- 15 ISO 2788: 1986 Documentation — Guidelines for the establishment and development of monolingual thesauri.
- 16 Kolodner J. L. (1993). Case-Based Reasoning. Los Altos, CA: Morgan Kaufmann.
- 17 Krenn B. Empirical implications on lexical association measures //Proceedings of the Ninth EURALEX International Congress, EURALEX 2000: Stuttgart, Germany, August 8th-12th, 2000. – 2000. – С. 359-371.
- 18 Lee C.S., Jian Z.-W., Huang L.-K., A fuzzy ontology and its application to news summarization // IEEE Transactions on Systems, Man and Cybernetics, PartB Vol. 5.– 2005. – p. 859–880.
- 19 Liu Y. et al. Text mining functional keywords associated with genes //Proc. Medinfo. – 2004. – С. 292-296.
- 20 Moshkin V.S., Filippov A.A., Yarushkina N.G. Integration of an applied ontology and wiki-resources in the context of the unified knowledge base// Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2017): материалы VII Междунар. научн.техн. конф. (Минск, 16-18 февраля 2017 г.) / редкол.: В. В. Голенков (отв. ред.) [и др.]. – Минск: БГУИР. – 2017. – С. 107–110.
- 21 Namestnikov A., Yarushkina N. Efficiency of Genetic algorithms for automated design problems // Известия Российской академии наук. Теория и системы

- управления. — 2002. — № 2. — С. 127–133.
- 22 Park Han-Saem , Cho Sung-Bae. A Fuzzy Rule-Based System with Ontology for Summarization of Multicamera Event Sequences. L. Rutkowski et al. (Eds.): ICAISC 2008, LNAI 5097, pp. 850–860, SpringerVerlag Berlin Heidelberg.
 - 23 Pazienza M. T., Pennacchiotti M., Zanzotto F. M. Terminology extraction: an analysis of linguistic and statistical approaches //Knowledge Mining. – Springer Berlin Heidelberg, 2005. – С. 255-279.
 - 24 Rajaraman A., Ullman J. D. Mining of massive datasets. – Cambridge University Press, 2011.
 - 25 Robertson S. Understanding inverse document frequency: on theoretical arguments for IDF //Journal of documentation. – 2004. – Т. 60. – №. 5. – С. 503-520.
 - 26 Sankar K. Pal, Simon C. K. Shiu. Foundations of Soft Case-Based Reasoning. New Jersey: Wiley, 2004, ISBN: 978-0-471-64466-8.
 - 27 Sosnin P. Pseudo-code programming of designer activities in development of software intensive systems Lecture Notes in Computer Science. 2012. Т. 7345 LNAI. С. 457-466.
 - 28 Sosnin P.I. Question-answer models of decision-making tasks in automated designing В сборнике: Proceedings - 22nd European Conference on Modelling and Simulation, ECMS 2008 2008. С. 173-180.
 - 29 Straccia U. Towards a Fuzzy Description Logic for the Semantic Web // 2nd European Semantic Web Conference, 2005. — P. 167-181.
 - 30 SWRL: A Semantic Web Rule Language Combining OWL and RuleML [Электронный ресурс]. — Режим доступа: <https://www.w3.org/Submission/SWRL> (дата обращения: 11.05.2016).
 - 31 The Protégé Ontology Editor and Knowledge Acquisition System [Электронный ресурс]. — Режим доступа: <http://protege.stanford.edu> (дата обращения: 12.08.2016).
 - 32 Trausan-Matu, Stefan. A Framework for an Ontology-Based Information System for Competence Management. Economy Informatics, 1-4/2008, p.105.

- 33 Valderrábanos S., Belskis A., Iraola L. TExtractor: a multilingual terminology extraction tool. Proceedings of HLT 2002, Second International Conference on Human Language Technology Research, M. Marcus, ed., Morgan Kaufmann, San Francisco, 2002.
- 34 Yaguo Lei. The Hybrid Intelligent Method Based on Fuzzy Inference System and Its Application to Fault Diagnosis, Fuzzy Inference System Theory and Applications, Dr. Mohammad Fazle Azeem (Ed.), 2012. – pp. 153-170.
- 35 Yarushkina N. Soft computing and complex system analysis // International Journal of General Systems. — 2001. — Vol. 30, № 1. — pp. 71–88.
- 36 Yarushkina N., Moshkin V., Klein V., Andreev I, Beksaeva E. Hybridization of Fuzzy Inference and Self-learning Fuzzy Ontology-Based Semantic Data Analysis. In: Proceedings of the First International Scientific Conference “Intelligent Information Technologies for Industry” (IITI’16), pp. 277–285 (2016)
- 37 Zadeh L. Fuzzy sets // Information and Control. — 1965. — №8. — P. 338-353.
- 38 Zhang T. Predictive Methods for Text Mining //Machine Learning Summer School-2006, Taipei. Available: videlectures.net/mlss06tw_zhang_pmtm. – 2006.
- 39 Zhou Z., Wang Y., Gu J. Markov-Based Automatic Term Extraction //Future BioMedical Information Engineering, 2008. FBIE'08. International Seminar on. – IEEE, 2008. – C. 86-89.
- 40 Zong-Yong Li, Zhi-xu Wang, Ai-hui Zhang, Yong Xu. The Domain Ontology and Domain Rules Based Requirements Model Checking. 2007.
- 41 Абрамян Т.С., Коровин М.Д., Боргест Н.М. Разработка онтологической модели оценки остаточного ресурса агрегатов авиационной техники // В сборнике: Information Technologies for Intelligent Decision Making Support (ITIDS'2015) Proceedings of the 3rd International Conference. 2015. С. 116-119.
- 42 Абу Д. М. Исследование и разработка системы формирования и реконфигурации архитектуры конструкторских САПР радиоэлектронной аппаратуры : автореф. дис. канд. техн. наук / Д. М. Абу. – Санкт-Петербург, 2009. – С. 16.

- 43 Аверкин А. Н., Батыршин И. З. и др. Нечеткие множества в моделях управления искусственного интеллекта / Под ред. Д. А. Поспелова. М.: Наука, 1986.
- 44 Авлукова Ю.Ф. Основы автоматизированного проектирования: учеб. пособие / Ю.Ф. Авлукова. – Минск: Выш. шк., 2013. – 217 с.: ил.
- 45 Андреев В.В., Батищев С.В., Виттих В.А., Ивкушкин К.В., Минаков И.А., Ржевский Г.А., Сафронов А.В., Скобелев П.О. Методы и средства создания открытых мультиагентных систем для поддержки процессов принятия решений Известия Российской академии наук. Теория и системы управления. 2003. № 1. С. 126-137.
- 46 Андреев И.А., Башаев В.А., Клейн В.В. Мошкин В.С. Определение вероятности терминологичности словоупотреблений в текстах конкретной предметной области // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов VIII-й Международной научно-практической конференции (Коломна, 18-20 мая 2015 г.). В 2-х томах. Т2. – М.: Физматлит, 2015. – С. 764–773.
- 47 Андреев И.А., Башаев В.А., Клейн В.В. Разработка программного средства для извлечения терминологии из текста на основании морфологических признаков, определяемых программой Mystem // Интегрированные модели и мягкие вычисления в искусственном интеллекте. — М. : Физматлит, 2013. — С. 1227–1236.
- 48 Андреев И.А., Башаев В.А., Клейн В.В., Мошкин В.С., Ярушкина Н.Г. Семантическая метрика терминологичности на основе онтологии предметной области // Автоматизация процессов управления. – 2014. – № 4 (38). – С. 76–84.
- 49 Андреев И.А., Башаев В.А., Клейн В.В., Мошкин В.С., Ярушкина Н.Г. Оценка терминологичности лексических единиц на основе онтологии предметной области // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2015): материалы V Междунар. научн.техн. конф. (Минск, 19-21 февраля 2015 г.) / редкол.: В. В. Голенков

- (отв. ред.) [и др.]. – Минск: БГУИР. – 2015.С. 395–400.
- 50** Андреев И.А., Бексаева Е.А., Клейн В.В., Мошкин В.С., Серков И.П. Лингвистический подход к автоматизированному построению предметной онтологии // Прикладные информационные системы: третья Всероссийская НПК (г. Ульяновск, 30 мая – 12 июня 2016 г.): сборник научных трудов / под ред. Е. Н. Эгова. – Ульяновск : УлГТУ, 2016. – С.256–263.
- 51** Анисимов Д.Н. Нечеткие алгоритмы управления: Уч. пособие. — М.: Изд-во МЭИ, 2004. — 61 с.
- 52** Армер А.И., Мошкин В.С. Подход к формированию наборов эталонов речевых команд с использованием онтологии // Онтология проектирования. – 2016. – Т.6, №3(21). – С.270-277.
- 53** Армер А.И., Мошкин В.С., Ярушкина Н.Г. Использование онтологии для формирования наборов эталонов речевых команд в задаче распознавания речевых команд на фоне шумов// Радиотехника. – 2016. – № 9. – С. 72–76.
- 54** Афанасьева Т.В., Ярушкина Н.Г. Нечеткий динамический процесс с нечеткими тенденциями в анализе временных рядов // Вестник Ростовского государственного университета путей сообщения. — 2011. Т. 3. — С. 7–16.
- 55** Барабанова Е.Ю., Башаев В.А., Клейн В.В., Мошкин В.С. Построение информационной поддержки автоматизированного проектирования управляющих программ для станков с ЧПУ // Радиотехника. – 2015. – № 6. – С. 63–67.
- 56** Башаев В.А., Мошкин В.С., Андреев И.А., Клейн В.В. Использование семантической метрики для решения задачи извлечения терминологии из текста проблемной области // Информатика и вычислительная техника: сборник научных трудов 6-й Всероссийской научно-технической конференции аспирантов, студентов и молодых ученых ИВТ-2014 / под ред. Н. Н. Войта. – Ульяновск: УлГТУ, 2014. – С.7278.
- 57** Береза А.Н., Курейчик В.М. Эволюционные методы синтеза структурных схемотехнических решений для проектирования информационно-телекоммуникационных систем // Известия Российской академии наук.

Теория и системы управления. 2006. № 2. С. 73-83.

- 58** Благодаров А.В. Алгоритмы категорирования персональных данных для систем автоматизированного проектирования баз данных информационных систем. / А.В. Благодаров, В.С. Зияутдинов, П.А. Корнев, В.Н. Малыш. — М. : Горячая линия-Телеком, 2013. — 116 с.
- 59** Боргест Н.М., Шустова Д.В., Гиматдинова С.Р. Иерархические и ассоциативные связи между терминами в тезаурусе на примере словаря проектанта // Вестник Самарского государственного аэрокосмического университета им. академика С.П. Королёва (национального исследовательского университета). 2012. № 2 (33). С. 228-236.
- 60** Боргест, Н.М. Тезаурус как основа онтологии концептуального проектирования самолета [Текст] / Н.М. Боргест, Д.В. Шустова // Труды Международной конференции с элементами научной школы для молодежи: сб. науч. тр.; Самарский гос. аэрокосм. ун-т им. Академика С.П. Королёва (национальный исследовательский ун-т). — Самара: Изд-во Самар.гос. аэрокосм. ун-та, 2010, — С. 353-356.
- 61** Борисов В. В., Круглов В. В., Федулов А. С. Нечеткие модели и сети. М.: Горячая линия-Телеком, 2007. 284 с.
- 62** Бочарников В.П. Fuzzy-технология: Математические основы. Практика моделирования в экономике / В. П. Бочарников. — Санкт-Петербург: «Наука» РАН, 2001. — 328 с.
- 63** Браславский П. И., Соколов Е. А. Автоматическое извлечение терминологии с использованием поисковых машин Интернета //Компьютерная лингвистика и интеллектуальные технологии. Тр. Международной конференции “Диалог. — 2007. — С. 89-94.
- 64** Браславский П. И., Соколов Е. А. Сравнение четырех методов автоматического извлечения двухсловных терминов из текста //Компьютерная лингвистика и интеллектуальные технологии. Тр. Международной конференции “Диалог. — 2006. — С. 88-94.
- 65** Браславский П., Соколов Е. Сравнение пяти методов извлечения терминов

- произвольной длины //Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» (Бекасово, 4–8 июня 2008). – 2008. – №. 7. – С. 14.
- 66** Бусленко И.В., Калашников В.В., Коваленко И.Н. Лекции по теории сложных систем. -М.: Сов. радио, 1975.
- 67** Вагин В. Н. Дедукция и обобщение в системах принятия решений / В. Н. Вагин. М. : Наука, гл. ред. физ.-мат. лит. 1988. 384 с.
- 68** Вагин В.Н., Михайлов И.С. Разработка метода интеграции информационных систем на основе метамоделирования и онтологии предметной области // Программные продукты и системы. – 2008. – С. 22-26.
- 69** Виттих В.А., Ситников П.В., Смирнов С.В. Онтологический подход к построению информационно-логических моделей в процессах управления социальными системами // Вестник компьютерных и информационных технологий. 2009. №5. С. 45-53.
- 70** Воронина И.Е., Пигалкова Е.А. Интеграция знаний продукционного характера в правовую онтологическую модель с помощью SWRL-правил // Вестник ВГУ, серия «Системный анализ и информационные технологии». – 2010. – № 2. – С. 139–144
- 71** Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. Учебник. – СПб.: Питер, 2000. – 384 с.
- 72** Гаврилова Т.А. Извлечение знаний: лингвистический аспект // Enterprise Partner. – 2001. – № 1 (27). – С. 24-28.
- 73** Глибовец Н. Н., Красиков Д. С. Интеллектуализация экспертных систем с помощью онтологий // Information science and computing. Book 28. «Problems of Computer Intellectualization», 2012. 84-90.
- 74** ГОСТ 7.24-2007. Система стандартов по информации, библиотечному и издательскому делу. Тезаурус информационно-поисковый многоязычный. Состав структура и основные требования к построению. – М. : Стандартинформ, 2010. – [Электронный ресурс]. — Режим доступа: <http://protect.gost.ru/document.aspx?control=7&id=128610> (дата обращения:

20.10.2016).

- 75** ГОСТ 7.24-90. Система стандартов по информации, библиотечному и издательскому делу. Тезаурус информационно-поисковый одноязычный. Правила разработки, структура, состав и форма представления.
- 76** ГОСТ 22487-77. Проектирование автоматизированное. Термины и определения.
- 77** Гуськов Г.Ю., Мошкин В.С., Наместников А.М., Филиппов А.А., Ярушкина Н.Г. Разработка многоагентной системы извлечения знаний из гетерогенных источников// Радиотехника. – 2016. – № 9. – С. 57–63.
- 78** Дворянкин А. М, Сипливая М. Б., Жукова И. Г. Интеграция рассуждений по прецедентам и онтологии в интеллектуальной системе поддержки инженерного анализа в области контактной механики // Известия Волгоградского государственного технического университета,. Волгоград. – 2008. -№4. - Том 2.
- 79** Дж.Парк, С. Маккей. Сбор данных в системах контроля и управления. Практическое руководство: – М.: ООО «Группа ИДТ», 2006. –504 с., илл., табл.
- 80** Добров Б.В., Лукашевич Н.В., Сыромятников С.В. Формирование базы терминологических словосочетаний по текстам предметной области // Тр. 5-й Всеросс. науч. конф. «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL2003). — СПб., 2003. — С. 201–210.
- 81** Емельянов В.В., Курейчик В.В., Курейчик В.М. теория и практика эволюционного моделирования Москва, 2003.
- 82** Заболеева-Зотова А. В., Камаев В. А. Лингвистическое обеспечение автоматизированных систем //Высшая школа. – 2008.
- 83** Загорулько Г.Б., Загорулько Ю.А. Онтологический подход к созданию системы поддержки принятия решений при остеопорозе // Информационный бюллетень рабочего семинара «Наукоемкое программное обеспечение», – С.48-51, – Новосибирск: ИСИ СО РАН, 2009, – 128с.

- 84 Загорулько Ю.А., Загорулько Г.Б., Шестаков В.К., Кононенко И.С. Концепция и архитектура тематического интеллектуального научного интернет-ресурса // Электронные библиотеки: перспективные методы и технологии, электронные коллекции / Труды XV Всероссийской научной конференции RCDL'2013. 2013. С. 57-62.
- 85 Заде Л. Понятие лингвистической переменной и ее применение к принятию приближенных решений. — М.: Мир, 1976. — 167 с.
- 86 Захаров В. П., Хохлова М. В. Анализ эффективности статистических методов выявления коллокаций в текстах на русском языке. — 2010.
- 87 Зубкова Т.М., Мустюков Н.А., Колобов А.Н. Реконфигурирование САПР для проектирования одношнековых экструдеров на основе модели нечеткого вывода Мамдани // Вестник Оренбургского государственного университета. 2013. № 1 (150). С. 176-181.
- 88 Интегральный метод принятия решений и анализа нечетких временных рядов / В. Новак [и др.] // Программные продукты и системы. 2008. — № 4. — С. 18.
- 89 Ким Д. П. Теория автоматического управления. Т. 2. Многомерные, нелинейные, оптимальные и адаптивные системы: Учеб. пособие. М.: Физматлит, 2004. 464 с.
- 90 Клейн В.В., Мошкин В.С., Ярушкина Н.Г. Разработка семантической метрики терминологичности лексических единиц в решении задачи формирования словарей // Прикладные информационные системы: вторая Всероссийская НПК (г. Ульяновск, 25 мая–7 июня 2015 г.): сборник научных трудов / под ред. Е.Н. Эгова. — Ульяновск: УлГТУ, 2015. — С. 199–206.
- 91 Круглов В.В., Дли М.И., Голунов Р.Ю. Нечеткая логика и искусственные нейронные сети. М.: Физматлит, 2001. 221 с.
- 92 Курейчик В.М., Медведева И.Б. Построение онтологий систем автоматизированного проектирования // В сборнике: Информационные технологии и математическое моделирование систем 2015 Труды

- международной научно-технической конференции. 2015. С. 29-30.
- 93** Кучеренко Е.И., Павлов Д.А. Некоторые аспекты анализа развития нечетких онтологий. // Искусственный интеллект. Донецк. 2005. — С.162-169.
- 94** Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzy TECH. – Санкт-Петербург: Издательство «БХВ-Петербург», 2005. 736 с.: ил.
- 95** Лесных А.Н., Ловчиков А.Н., Лукьяненко М.В. Методы моделирования и синтеза систем электропитания космических аппаратов // Монография / . Красноярск, 2015.
- 96** Лисецкий Ю.М. Сложные системы // Программные продукты и системы. 2005. №3.С.1
- 97** Лукашевич Н. В., Логачев М. Ю. Использование методов машинного обучения для извлечения слов-терминов // Компьютерная лингвистика и семантический Web: по материалам двенадцатой национальной конференции по искусственному интеллекту КИИ-2010
- 98** Лукашевич Н. В., Логачев Ю. М. Комбинирование признаков для автоматического извлечения терминов //вычислительные методы и программирование. – 2010. – Т. 11. – С. 109.
- 99** Лукашевич Н.В. Тезаурусы в задачах информационного поиска // М.: Издательство МГУ, 2011. - 512 с.
- 100** Мелихов А.Н., Берштейн Л.С., Коровин С.Я. Ситуационные советующие системы с нечеткой логикой – Наука — М. : Физматлит, 2010. 272 с.
- 101** Митрофанова О.А., Константинова Н.С. Онтологии как системы хранения знаний // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению «Информационно-телекоммуникационные системы». — 2008. — С. 54
- 102** Мошкин В.С. Интеграция знаний продукционного характера в онтологическую модель предметной области//Вузовская наука в современных условиях: сборник материалов 48-й научно-технической конференции. Ч.2. – Ульяновск: УлГТУ, 2014. – С. 164–167.
- 103** Мошкин В.С. Использование резонеров в процессе резюмирования

- временных рядов на основе онтологии проблемной области // Информатика и вычислительная техника: сборник научных трудов 5-й Всероссийской научно-технической конференции аспирантов, студентов и молодых ученых ИВТ-2013 / под ред. Н. Н. Войта. – Ульяновск: УлГТУ, 2013. – С.126-129.
- 104** Мошкин В.С. Онтологический анализ временных рядов//Вузовская наука в современных условиях: сборник материалов 47-й научно-технической конференции. Ч.2. – Ульяновск: УлГТУ, 2013. – С. 272-275.
- 105** Мошкин В.С., Андреев И.А., Башаев В.А., Клейн В.В. Семантическая метрика «термин/не термин» на основе онтологии проблемной области // Методы и технологии гибридного и синергетического искусственного интеллекта: материалы I международной Пospelовской летней школы-семинара для студентов, магистрантов и аспирантов. Калининград: Изд-во БФУ им. И. Канта, 2014. – С. 67–73.
- 106** Мошкин В.С., Пирогов А.Н., Тимина И.А., Шишкин В.В., Ярушкина Н.Г. Интеллектуальный анализ проектных и терминологических метрик в управлении проектами // Автоматизация процессов управления. – 2016. – № 4 (46). – С. 84–91.
- 107** Мошкин В.С., Филиппов А.А., Ярушкина Н.Г. Разработка многоагентной базы знаний в рамках технологической платформы Athene // Пятнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ – 2016: Труды конференции. Т.3. – Смоленск: Универсум, 2016. – С. 261-269.
- 108** Мошкин В.С., Ярушкина Н.Г. Взаимодействие онтологического анализа и механизмов логического вывода // Гибридные и синергетические интеллектуальные системы: материалы II международного Пospelовского симпозиума / под ред. Д-ра техн. Наук, проф. А.В. Колесникова. – Калининград: Изд-во БФУ им. И.Канта, 2014 С.239–245.
- 109** Мошкин В.С., Ярушкина Н.Г. Гибридизация алгоритмов извлечения знаний из текстов и механизма прецедентов в процессе расширения онтологии// Пятнадцатая национальная конференция по искусственному интеллекту с

- международным участием КИИ – 2016: Труды конференции. Т.1. – Смоленск: Универсум, 2016. – С. 80-87.
- 110** Мошкин В.С., Ярушкина Н.Г. Интеграция нечетких онтологий и базы прецедентов в единой семантической среде // В сборнике: Гибридные и синергетические интеллектуальные системы Материалы III Всероссийской Поспеловской конференции с международным участием. Под редакцией А. В. Колесникова. 2016. С. 411-419.
- 111** Мошкин В.С., Ярушкина Н.Г. Логический вывод на основе нечетких онтологий // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов VIII-й Международной научно-практической конференции (Коломна, 18-20 мая 2015 г.). В 2-х томах. Т1. – М.: Физматлит, 2015. – С. 259–267.
- 112** Мошкин В.С., Ярушкина Н.Г. Методики построения нечетких онтологий сложных предметных областей // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2015): материалы V Междунар. научн.техн. конф. (Минск, 19-21 февраля 2015 г.) / редкол.: В. В. Голенков (отв. ред.) [и др.]. – Минск: БГУИР.– 2015. – С. 401–406.
- 113** Мошкин В.С., Ярушкина Н.Г. Модифицированный метод логического вывода знаний на основе нечеткой онтологии и базы прецедентов // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2016): материалы VI Междунар. научн.техн. конф. (Минск, 18-20 февраля 2016 г.) / редкол.: В. В. Голенков (отв. ред.) [и др.]. – Минск: БГУИР. – 2016. – С. 265–270.
- 114** Мошкин В.С., Ярушкина Н.Г. Особенности интеграции механизмов логического вывода в онтологическую модель представления знаний с помощью SWRL-правил // Четырнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ – 2014: Труды конференции. Т.1. – Казань: Изд-во РИЦ «Школа». – 2014. – С. 173-181.
- 115** Мошкин В.С., Ярушкина Н.Г. Построение нечеткой онтологии тестирования состояния ЛВС группой экспертов // Шестая международная конференция

- «Системный анализ и информационные технологии» САИТ–2015 (15–20 июня 2015 г., г. Светлогорск, Россия): Труды конференции. В 2-х т. – Т. 2. М.: – С. 107–111.
- 116** Мошкин В.С., Ярушкина Н.Г. Система онтологического анализа временных рядов // Автоматизация процессов управления. – 2014. – № 2 (36). – С. 78-85.
- 117** Нечеткие гибридные системы / И.З.Батыршин, А.О.Недосекин, А.А.Стецко, В.Б.Тарасов, А.В.Язенин, Н.Г.Ярушкина. — М.: Физматлит, 2007.
- 118** Орлова А.А., Боргест Н.М. Разработка руководств по эксплуатации с помощью онтологий // В сборнике: Information Technologies for Intelligent Decision Making Support (ITIDS'2016) Proceedings of the 4th International Conference. 2016. С. 1-5.
- 119** Питер Джексон. Введение в экспертные системы = Introduction to Expert Systems. – 3-е изд. – М.: Вильямс, 2001. – 624 с.
- 120** Разин В.В., Тузовский А.Ф. Метод принятия решений на основе анализа ситуации и семантических технологий // Известия Томского политехнического университета – Томск – 2012. – №5, С. 188-193.
- 121** Смирнов С.В. Онтологический анализ предметных областей моделирования // Известия Самарского научного центра Российской академии наук. 2001. Т. 3. № 1. С. 62-70.
- 122** Смирнов С.В. Онтологический подход к формированию гетерогенных сред моделирования // Вестник Самарского государственного технического университета. Серия: Технические науки. 2011. № 4 (32). С. 50-61.
- 123** Смирнов С.В. Онтологическое моделирование в ситуационном управлении // Онтология проектирования. 2012. № 2. С. 16-24.
- 124** Соловьев В.Д., Добров Б.В., Иванов В.В., Лукашевич Н.В. Онтологии и тезаурусы. Учебное пособие. – М. : 2006.
- 125** Соснин П.И. Вопросно-ответное моделирование в разработке автоматизированных систем // монография / П. И. Соснин ; Федеральное агентство по образованию, Ульяновский гос. технический ун-т. Ульяновск, 2007.

- 126** Тарасов В.Б., Калуцкая А.П., Святкина М.Н. Гранулярные, нечеткие и лингвистические онтологии для Обеспечения взаимопонимания между когнитивными агентами // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2012): материалы II междунар. науч.-техн. конф. — Минск: БГУИР, 2012. — С. 267-278.
- 127** Трахтенгерц Э.А. Компьютерная поддержка принятия решений: Научно-практическое издание. Серия XXI века. — М.: СИНТЕГ, 2012. 396 с.
- 128** Филиппов А.А., Мошкин В.С., Шалаев Д.О., Ярушкина Н.Г. Единая онтологическая платформа интеллектуального анализа данных // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2016): материалы VI Междунар. научн.техн. конф. (Минск, 18-20 февраля 2016 г.) / редкол.: В. В. Голенков (отв. ред.) [и др.]. — Минск: БГУИР. — 2016. — С. 77–82.
- 129** Филиппов А.А., Мошкин В.С., Ярушкина Н.Г., Шалаев Д.О. Назработка агента для управления знаниями с использованием графовой базы данных // В сборнике: Гибридные и синергетические интеллектуальные системы Материалы III Всероссийской Поспеловской конференции с международным участием. Под редакцией А. В. Колесникова. 2016. С. 360-367.
- 130** Хорошевский В.Ф. Обработка естественно-языковых текстов: от моделей понимания языка к технологиям извлечения знаний // Новости искусственного интеллекта. — 2002. — № 6. С. 19–26.
- 131** Частиков А.П., Гаврилова Т.А., Белов Д.Л. Разработка экспертных систем. Среда CLIPS. — СПб.: БХВ-Петербург, 2003. — 393 с.: ил.
- 132** Ягунова Е. В., Пивоварова Л. М. Природа коллокаций в русском языке. Опыт автоматического извлечения и классификации на материале новостных текстов //НТИ.—Сер. — 2010. — Т. 2. — С. 30-40.
- 133** Ярушкина Н.Г. Основы теории нечетких и гибридных систем: Учеб.пособие. — М. : Финансы и статистика, 2004.
- 134** Ярушкина Н.Г., Афанасьева Т.В. Нечеткие временные ряды как инструмент

- для оценки и измерения динамики процессов // Датчики и системы. — 2007. — № 12. — С. 46–50.
- 135** Ярушкина Н.Г., Афанасьева Т.В., Перфильева И.Г. Интеллектуальный анализ временных рядов. Учебное пособие. — Ульяновск.— 2010.
- 136** Ярушкина Н.Г., Вельмисов А.П., Стецко А.А. Средства data minig для нечетких реляционных серверов данных // Информационные технологии. — 2007. — № 6. — С. 20–29.
- 137** Ярушкина Н.Г., Мошкин В.С. Применение онтологического подхода к анализу состояния локальной вычислительной сети // Радиотехника. 2014. — № 7. — С. 120–124.
- 138** Ярушкина Н.Г., Мошкин В.С. Онтологический подход к анализу временных рядов // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сб. научных трудов VII-й Международной научно-практической конференции. — Коломна – 2013. — № 2. — С. 529–537
- 139** Ярушкина Н.Г., Мошкин В.С. Онтология как основа семантического анализа данных в задачах автоматизированного проектирования // Нечеткие системы и мягкие вычисления (НСМВ-2014): труды Шестой всероссийской научно – практической конференции. – Т.1. –СПб.: Политехника-сервис, 2014. – С. 174-182.
- 140** Ярушкина Н.Г., Мошкин В.С. Подход к обучению онтологии на основе гибридизации алгоритмов извлечения знаний из текстов и механизма прецедентов // Вестник Ростовского государственного университета путей сообщения. 2016. № 2 (62). С. 78-83.
- 141** Ярушкина Н.Г., Мошкин В.С. Применение алгоритма логического вывода на основе FuzzyOWL-онтологии // Радиотехника. – 2015. – № 6. – С. 68–72.

**ПРИЛОЖЕНИЕ А. СВИДЕТЕЛЬСТВА О ГОСУДАРСТВЕННОЙ
РЕГИСТРАЦИИ ПРОГРАММ ДЛЯ ЭВМ**

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2014616701

**Онтологически-ориентированная система логического
вывода**

Правообладатель: *федеральное государственное бюджетное
образовательное учреждение высшего профессионального
образования «Ульяновский государственный технический
университет» (RU)*

Авторы: *Ярушкина Надежда Глебовна (RU),
Мошкин Вадим Сергеевич (RU)*



Заявка № **2014614071**

Дата поступления **05 мая 2014 г.**

Дата государственной регистрации
в Реестре программ для ЭВМ **02 июля 2014 г.**

Руководитель Федеральной службы
по интеллектуальной собственности

Б.П. Симонов

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2014616248

Онтологически-ориентированная система извлечения терминологии

Правообладатель: *федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Ульяновский государственный технический университет» (RU)*

Авторы: *Ярушкина Надежда Глебовна (RU), Башаев Виталий Александрович (RU), Мошкин Вадим Сергеевич (RU), Клейн Виктор Викторович (RU), Андреев Илья Алексеевич (RU)*

Заявка № 2014613574

Дата поступления 21 апреля 2014 г.

Дата государственной регистрации
в Реестре программ для ЭВМ 18 июня 2014 г.



Руководитель Федеральной службы
по интеллектуальной собственности

Б.П. Симонов

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2017611364

**Интеллектуальная экспертная система на основе нечеткой
онтологии**

Правообладатель: *федеральное государственное бюджетное
образовательное учреждение высшего образования «Ульяновский
государственный технический университет» (RU)*

Авторы: *Ярушкина Надежда Глебовна (RU),
Мошкин Вадим Сергеевич (RU)*

Заявка № 2016660651

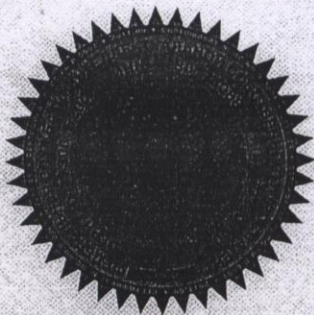
Дата поступления 11 октября 2016 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 02 февраля 2017 г.

Руководитель Федеральной службы
по интеллектуальной собственности

Г.П. Ивлиев



ПРИЛОЖЕНИЕ Б. АКТ ВНЕДРЕНИЯ

УТВЕРЖДАЮ

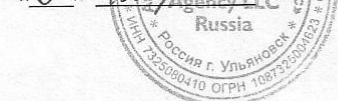
Директор

ООО «Агентство

Переводов «Юнитранс»

В.А. Башаев

« 9 » июня 2017 г.



А К Т

об использовании результатов кандидатской диссертации В.С. Мошкина
“Исследование представления терминологии в лингвистическом обеспечении
САПР на основе интеграции нечетких онтологий и логического вывода”

Настоящим актом подтверждается использование следующих научных и практических результатов диссертационной работы В.С. Мошкина “Исследование представления терминологии в лингвистическом обеспечении САПР на основе интеграции нечетких онтологий и логического вывода” для формирования базы отраслевых знаний «Юнитех» с целью краткосрочной отраслевой подготовки специалистов по мультязычной проектной документации:

- онтологически-ориентированная методика извлечения терминологии из проектных документов с использованием базового ядра предметной онтологии;
- алгоритм расширения онтологического представления терминологии предметной области технической системы с использованием базового ядра онтологии интеллектуальной САПР посредством извлечения терминологии из проектных документов;
- программная система поддержки проектирования, являющаяся компонентой интеллектуальной САПР для представления терминологии лингвистического обеспечения.

Эффективность использования научно-технических результатов подтверждена экспериментальными исследованиями, целью которых являлось оценка временных затрат на отраслевую подготовку специалистов по работе с мультязычной проектной документацией с использованием учебной базы отраслевых знаний «Юнитех», полученной с применением предложенных

алгоритмов и методик.

Для автоматизации процесса формирования учебной базы знаний «Юнитех» сотрудниками ООО Агентство переводов «Юнитранс» была разработана базовая OWL-онтология на основе руководства по эксплуатации токарно-фрезерного станка с числовым программным управлением, а также применены предлагаемые в рамках данного диссертационного исследования алгоритмы и методики извлечения терминологии из проектных документов с использованием базового ядра предметной онтологии.

В течение 6 месяцев специалисты изучали базу знаний «Юнитех». «Учебный справочник технического переводчика: станки с ЧПУ», эксплуатационную документацию, а также выполняли рабочие переводы. В начале стажировки и по прошествии каждого месяца обучения стажеры выполняли перевод контрольных образцов проектной документации, что позволило в динамике отследить показатели качества перевода.

Как показали результаты проводимых экспериментов, использование базы знаний «Юнитех», полученной путем извлечения терминологии предложенными в диссертационной работе алгоритмами, в процессе обучения специалистов по работе с мультязычной проектной документацией позволяет обеспечить специалиста 78% приобретаемых отраслевых знаний за первые два месяца обучения, тем самым сократив срок базового обучения с 6 до 2-х месяцев.

Директор ООО «Агентство переводов «Юнитранс»

В.А. Башаев



**ПРИЛОЖЕНИЕ В. ДИПЛОМ ПОБЕДИТЕЛЯ КОНКУРСА УМНИК В
ОБЛАСТИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ПРИ ПОДДЕРЖКЕ
ФОНДА СОДЕЙСТВИЯ РАЗВИТИЮ МФП В НАУЧНО-ТЕХНИЧЕСКОЙ
СФЕРЕ.**



**ПРИЛОЖЕНИЕ Г. ТАБЛИЦА ВЫВОДА РЕКОМЕНДАЦИЙ ПРИ
ПРОВЕДЕНИИ ЭКСПЕРИМЕНТОВ ПО ОЦЕНКИ СОСТОЯНИЯ ЛВС**

№	Моделируемые проблемы	Вывод системы	Степень уверенности
1	Быстрый рост фоновой нагрузки, рост времени реакции пользовательского ПО. Искусственное повышение трафика сети.	Возможная причина – перегруженность канала связи. Требуется изменение архитектуры сети. Число станций в перегруженных доменах следует уменьшить, а станции, создающие наибольшую	0,7
		Возможная причина – проблема в настройках сетевой платы сервера домена, проверьте целостность сетевой карты и правильность ее настройки.	0,4
		Возможная причина сбоев в подключении – заражение системы. Проверьте сервисы, загружающие сеть. Также желательна настройка брандмауэра или прокси агента для защиты сети.	0,2
2	Активация режима Half Duplex, рост фоновой нагрузки	Возможная причина – несоответствие характеристик сетевого коммутатора объему трафика. Требуется замена коммутатора или изменение его настроек.	0,9
		Возможная причина – проблема в настройках сетевой платы сервера домена, проверьте целостность сетевой карты и правильность ее настройки	0,75
3	Повреждение одного из коммутационных шнуров, рост числа коллизий	Возможная причина – неполадки с подключением коммутационных шнуров. Причина – неправильная организация заземления компьютеров, включенных в локальную сеть.	0,9
		Возможная причина сбоев в системе - проблема в СКС. Возможна проблема в распиновке кабелей, отходящих от сервера, проверьте правильность обжима.	0,4
		Возможная причина сбоев в системе – аппаратный сбой в работе узла сети. Требуется изменение архитектуры сети для повышения надежности передачи данных.	0,25

4	Наличие дефектов у портов коммутатора, рост числа ошибок CRC, рост фоновой нагрузки	Возможная причина – неполадки работы портов сетевого коммутатора или повреждение коммутационного шнура, ведущего к проблемному хосту	0,7
		Возможная проблема – сбой в питании сетевой платы клиента.	0,3
5	Наличие дефектов сетевой платы, быстрый рост числа локальных и удаленных коллизий	Возможная причина – проблема в настройках сетевой платы сервера домена, проверьте целостность сетевой карты и правильность ее настройки	0,7
		Возможная причина сбоев в системе – отсутствие нативных драйверов или использование универсальных аналогов. Переустановите драйвера сетевой платы сервера.	0,6
6	Наличие дефектов в кабеле, соединяющем сервер и клиента, нестабильная передача данных	Возможная причина – неисправность кабельной системы. Проверьте целостность кабельной системы.	0,9
		Возможная причина сбоев в системе - проблема в СКС. Возможна проблема в распиновке кабелей, отходящих от сервера, проверьте правильность обжима.	0,65
7	Программное отключение сетевой платы клиента	Возможная причина – неверные настройки BIOS. Убедитесь в том, что в настройках системы BIOS включена сетевая интерфейсная карта.	0,6
		Возможная причина сбоев в подключении – настройки сетевого экрана. Проверьте настройки файервола сети.	0,5
		Возможная причина сбоев в системе – отсутствие нативных драйверов или использование универсальных аналогов. Переустановите драйвера сетевой платы сервера.	0,5
8	Блокировка трафика брандмауэром сервера	Возможная причина сбоев в подключении – настройки сетевого экрана. Проверьте настройки файервола сети.	0,8

		Возможная причина сбоев в подключении – заражение системы. Проверьте сервисы, загружающие сеть. Также желательна настройка брандмауэра или прокси агента для защиты сети.	0,6
9	Быстрый рост фоновой нагрузки, рост времени реакции пользовательского ПО, рост нагрузки на все узлы сети.	Возможная причина сбоев в подключении – заражение системы. Проверьте сервисы, загружающие сеть. Также желательна настройка брандмауэра или прокси агента для защиты сети.	0,7
		Возможная причина – проблема в настройках сетевой платы сервера домена, проверьте целостность сетевой карты и правильность ее настройки.	0,3
10	Сбои в передаче данных, быстрый рост числа коллизий, переустановка драйверов сетевой карты сервера	Возможная причина сбоев в системе – отсутствие нативных драйверов или использование универсальных аналогов. Переустановите драйвера сетевой платы сервера.	0,8
		Возможная проблема – сбой в питании сетевой платы сервера.	0,1
11	Отсутствие связи с сервером, отсутствие механических повреждений СКС	Возможная проблема – сбой в питании сетевой платы сервера.	0,6
		Возможная причина сбоев в системе – отсутствие нативных драйверов или использование универсальных аналогов. Переустановите драйвера сетевой платы сервера.	0,5
		Возможная причина – неполадки работы портов сетевого коммутатора или повреждение коммутационного шнура, ведущего к проблемному хосту	0,3
12	Отсутствие связи с коммутатором, отсутствие механических повреждений СКС	Возможная проблема – сбой в питании сетевой платы клиента.	0,8
		Возможная причина – проблема в настройках сетевой платы сервера домена, проверьте целостность сетевой карты и правильность ее настройки	0,4

13	Частая самопроизвольная перезагрузка клиента, отсутствие механических повреждений СКС	Возможная проблема –неисправность сетевой платы клиента.	0,8
		Возможная причина сбоев в подключении – заражение системы. Проверьте сервисы, загружающие сеть. Также желательна настройка брандмауэра или прокси агента для защиты сети.	0,7
14	Ограничение доступа к определенным хостам сети, рост трафика незначительный	Возможная проблема –ошибка в работе веб-сервера. Проверьте правильность настройки и работы веб-сервера на машине-сервере.	0,9
		Возможная причина сбоев в подключении – настройки сетевого экрана. Проверьте настройки файервола сети.	0,6
		Возможная причина сбоев в подключении – заражение системы. Проверьте сервисы, загружающие сеть. Также желательна настройка брандмауэра или прокси агента для защиты сети.	0,4
15	Высокое значение затухания сигнала в сети	Возможная причина сбоев в системе - выход за пределы возможностей передачи сигнала внутри витой пары. Требуется коррекция архитектуры сети. Требуется дополнительный ретранслятор.	0,6
16	Отсутствие связи с подсетями, хорошая связь внутри сети	Возможная причина сбоев - ошибка маршрутизации. В настройках узла не указан стандартный шлюз (default gateway) или неверный IP-адрес стандартного шлюза.	0,7
		Возможная причина сбоев в системе – разные маски в настройках сети узлов.	0,35
17	Сбой в доставке пакетов между хостами	Возможная причина сбоев в системе - ошибка маршрутизации. Неверно настроено ICMP-перенаправление.	0,9
		Возможная причина сбоев в системе – разные маски в настройках сети узлов.	0,4
		Возможная причина сбоев в системе - неверные TCP/IP настройки хостов. Проверьте корректность настроек TCP/IP.	0,4

18	Узел успешно пересылает пакеты другим узлам в той же самой подсети, но не может	Возможная причина сбоев в системе - ошибка маршрутизации. Стандартный шлюз для этого узла находится в другой логической подсети, т.е. адрес или маска в настройках узла указана неправильно.	0,8
		Возможная проблема – некорректная работа DHCP-сервера. Проверьте правильность настроек сервера.	0,65
19	Часть узлов в какой-либо подсети может установить связь с узлами из других подсетей, а часть- нет.	Возможная причина сбоев в системе - ошибка маршрутизации. Проблема в настройке стандартного шлюза маршрутизатора, в котором для интерфейса локальной сети указана маска, отличающаяся от маски, заданной узлам.	0,6
		Возможная причина сбоев в системе - неверные TCP/IP настройки хостов. Проверьте корректность настроек TCP/IP.	0,3
20	Часть узлов в какой-либо сети VLAN может установить связь с узлами из	Возможная причина сбоев в системе – разные маски в настройках сети узлов.	0,8
		Возможная причина сбоев в системе - ошибка маршрутизации. Неверно настроено ICMP-перенаправление.	0,7
21	Потеря данных при прохождении через конкретный хост, отсутствие механических повреждений в СКС	Возможная причина сбоев в системе – аппаратный сбой в работе узла сети. Требуется изменение архитектуры сети для повышения надежности передачи данных.	0,9
		Возможная причина сбоев в подключении – настройки сетевого экрана. Проверьте настройки файервола сети.	0,8
		Возможная проблема – сбой в питании сетевой платы клиента.	0,8
22	Потери пакетов на конкретных участках, подходящих к серверу, стабильно высокий уровень потерь	Возможная причина сбоев в системе - проблема в СКС. Возможна проблема в распиновке кабелей, отходящих от сервера, проверьте правильность обжима.	0,7
		Возможная причина сбоев в системе - выход за пределы возможностей передачи сигнала внутри витой пары. Требуется коррекция архитектуры сети. Требуется дополнительный ретранслятор.	0,3

23	Потери пакетов на конкретных участках, стабильно высокий	Возможная причина сбоев в системе - проблема в СКС. Возможна проблема в распиновке кабелей, подходящих к хосту, проверьте правильность обжима.	0,7
24	Отсутствие механических повреждений, корректность работы сетевой платы, неверно	Возможная причина сбоев в системе - неверные TCP/IP настройки клиента. Проверьте корректность настроек TCP/IP.	0,7
25	Неверно введены все настройки маршрутизации	-	
26	Ошибки при регистрации доменных имен и распределении внутренних IP-адресов	Возможная проблема – некорректная работа DHCP-сервера. Проверьте правильность настроек сервера.	0,9
		Возможная причина сбоев в системе - неверные TCP/IP настройки клиента. Проверьте корректность настроек TCP/IP.	0,8
27	Сбои в передаче данных, медленный рост числа коллизий, переустановка драйверов сетевой карты сервера	Возможная причина сбоев в системе – отсутствие нативных драйверов или использование универсальных аналогов. Переустановите драйвера сетевой платы сервера.	0,8
28	Выборочное установление связи между узлами одной сети с узлами из другой.	Возможная причина сбоев в системе - ошибка маршрутизации. Проблема в настройке стандартного шлюза маршрутизатора, в котором для интерфейса локальной сети указана маска, отличающаяся от маски, заданной узлам.	0,6
		Возможная причина сбоев в системе - неверные TCP/IP настройки хостов. Проверьте корректность настроек TCP/IP.	0,5

29	Активация режима Full Duplex, быстрый рост фоновой нагрузки	Возможная причина – несоответствие характеристик сетевого коммутатора объему трафика. Требуется замена коммутатора или изменение его настроек.	0,9
		Возможная причина – проблема в настройках сетевой платы сервера домена, проверьте целостность сетевой карты и правильность ее	0,3
30	Самопроизвольная перезагрузка клиента, сбой в контакте с сервером	Возможная проблема –неисправность сетевой платы клиента.	0,8
		Возможная причина сбоев в подключении – заражение системы. Проверьте сервисы, загружающие сеть. Также желательна настройка	0,3

ПРИЛОЖЕНИЕ Д. ПРИМЕР ПРЕДСТАВЛЕНИЯ НЕЧЕТКОГО ПРАВИЛА В ФОРМАТЕ SWRL

```
<swrl:Imp rdf:about="file://OntoRules">
  <swrl:body>
    <swrl:AtomList>
      <rdf:rest>
        <swrl:AtomList>
          <rdf:rest>
            <swrl:AtomList>
              <rdf:rest>
                <swrl:AtomList>
                  <rdf:rest>
                    <swrl:AtomList>
                      <rdf:rest>
                        <swrl:AtomList>
                          <rdf:first>
                            <swrl:IndividualPropertyAtom>
                              <swrl:argument2>
                                <Тенденция rdf:ID="Рост"/>
                              </swrl:argument2>
                              <swrl:propertyPredicate
rdf:resource="#имеетТенденцию"/>
                                <swrl:argument1>
                                  <Параметры_работоспособности_сети
rdf:ID="Утилизация_канала_связи">
                                    <rdfs:label rdf:resource="#Утилизация_канала_связи"/>
                                  </Параметры_работоспособности_сети>
                                </swrl:argument1>
                              </swrl:IndividualPropertyAtom>
                            </rdf:first>
```

```

<rdf:rest>
  <swrl:AtomList>
    <rdf:rest>
      <swrl:AtomList>
        <rdf:rest>
          <swrl:AtomList>
            <rdf:first>
              <swrl:IndividualPropertyAtom>
                <swrl:propertyPredicate
rdf:resource="#имеетХарактер"/>
                <swrl:argument2>
                  <Характер_тенденции rdf:ID="Быстрый"/>
                </swrl:argument2>
                <swrl:argument1 rdf:resource="#Пост"/>
              </swrl:IndividualPropertyAtom>
            </rdf:first>
          <rdf:rest
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
        </swrl:AtomList>
      </rdf:rest>
    <rdf:first>
      <swrl:IndividualPropertyAtom>
        <swrl:argument2 rdf:resource="#Пост"/>
        <swrl:argument1>
          <Параметры_работоспособности_сети
rdf:ID="Число_коллизий_в_сети">
          <rdfs:label
rdf:resource="#Число_коллизий_в_сети"/>

```

```

        </Параметры_работоспособности_сети>
    </swrl:argument1>
    <swrl:propertyPredicate
rdf:resource="#имеетТенденцию"/>
    </swrl:IndividualPropertyAtom>
</rdf:first>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
    <swrl:IndividualPropertyAtom>
        <swrl:argument1>
            <swrl:Variable rdf:ID="x"/>
        </swrl:argument1>
        <swrl:argument2
rdf:resource="#Число_коллизий_в_сети"/>
        <swrl:propertyPredicate
rdf:resource="#включаетПоказатель"/>
        </swrl:IndividualPropertyAtom>
    </rdf:first>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
    <swrl:IndividualPropertyAtom>
        <swrl:argument1 rdf:resource="#Пост"/>
        <swrl:argument2>
            <Характер_тенденции rdf:ID="Постепенный"/>
        </swrl:argument2>
        <swrl:propertyPredicate
rdf:resource="#имеетХарактер"/>
    </swrl:IndividualPropertyAtom>

```



```

        </rdf:first>
        </swrl:AtomList>
    </rdf:rest>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
    <swrl:IndividualPropertyAtom>
        <swrl:propertyPredicate
rdf:resource="#включаетПоказатель"/>
        <swrl:argument2
rdf:resource="#Утилизация_канала_связи"/>
        <swrl:argument1 rdf:resource="#x"/>
    </swrl:IndividualPropertyAtom>
</rdf:first>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
    <swrl:IndividualPropertyAtom>
        <swrl:argument2 rdf:resource="#Постепенный"/>
        <swrl:propertyPredicate rdf:resource="#имеетХарактер"/>
        <swrl:argument1 rdf:resource="#Пост"/>
    </swrl:IndividualPropertyAtom>
</rdf:first>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
    <swrl:IndividualPropertyAtom>
        <swrl:argument1>
            <Параметры_работоспособности_сети
rdf:ID="Фоновая_нагрузка"/>

```

```

        </swrl:argument1>
        <swrl:propertyPredicate rdf:resource="#имеетТенденцию"/>
        <swrl:argument2 rdf:resource="#Пост"/>
    </swrl:IndividualPropertyAtom>
</rdf:first>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
    <swrl:IndividualPropertyAtom>
        <swrl:argument2 rdf:resource="#Фоновая_нагрузка"/>
        <swrl:argument1 rdf:resource="#x"/>
        <swrl:propertyPredicate rdf:resource="#включаетПоказатель"/>
    </swrl:IndividualPropertyAtom>
</rdf:first>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
    <swrl:ClassAtom>
        <swrl:argument1 rdf:resource="#x"/>
        <swrl:classPredicate rdf:resource="#Ситуация"/>
    </swrl:ClassAtom>
</rdf:first>
</swrl:AtomList>
</swrl:body>
<swrl:head>
    <swrl:AtomList>
        <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
        <rdf:first>
            <swrl:IndividualPropertyAtom>
                <swrl:propertyPredicate rdf:resource="#предполагает"/>

```

```

    <swrl:argument1 rdf:resource="#x"/>
    <swrl:argument2>
      <Рекомендации rdf:ID="Проблема_архитектуры">
        <имеетЗначение
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >Причина – перегруженность канала связи. Требуется изменение
архитектуры сети. Число станций в перегруженных доменах следует
уменьшить, а станции, создающие наибольшую загрузку, – подключить к
выделенным портам</имеетЗначение>
        </Рекомендации>
      </swrl:argument2>
    </swrl:IndividualPropertyAtom>
  </rdf:first>
</swrl:AtomList>
</swrl:head>
<swrla:isRuleEnabled
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >true</swrla:isRuleEnabled>
</swrl:Imp>...

```

ПРИЛОЖЕНИЕ Е. ФРАГМЕНТ FUZZYOWL-ОНТОЛОГИИ ОЦЕНКИ СОСТОЯНИЯ ЛВС

```
///Объявление FuzzyOWL-онтологии

<?xml version="1.0"?>
<!DOCTYPE Ontology [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.w3.org/TR/2003/PR-owl-guide-20031209/lan"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:swrl="http://swrl.stanford.edu/ontologies/built-ins/3.4/swrl.owl#"
  ontologyIRI="http://www.w3.org/TR/2003/PR-owl-guide-20031209/lan">
  <Prefix name="" IRI="http://www.w3.org/TR/2003/PR-owl-guide-20031209/lan#"/>
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
  <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
  <Prefix name="vin" IRI="http://www.w3.org/TR/2003/PR-owl-guide-20031209/lan#"/>
  <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
  <Prefix name="lan" IRI="http://www.w3.org/TR/2003/PR-owl-guide-20031209/lan#"/>
  <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
  <owl:Ontology rdf:about="file:/D:/LocalNetwork.owl">
  <owl:imports rdf:resource="http://swrl.stanford.edu/ontologies/3.3/swrla.owl"/>
```

```

    <owl:imports
        rdf:resource="http://sqwrl.stanford.edu/ontologies/built-
ins/3.4/sqwrl.owl"/>
</owl:Ontology>

<Annotation>
    <AnnotationProperty abbreviatedIRI="rdfs:comment"/>
    <Literal datatypeIRI="&rdf;PlainLiteral">Fuzzy OWL ontology of LAN</Literal>
</Annotation>

```

/// Определение типа используемой нечеткой логики

```

<Annotation>
    <AnnotationProperty IRI="#fuzzyLabel"/>
    <Literal datatypeIRI="&rdf;PlainLiteral"><fuzzyOwl2 fuzzyType="ontology">;
<FuzzyLogic logic="zadeh" />;
</fuzzyOwl2>;</Literal>
</Annotation>

```

/// Инициализация классов онтологии и отношения род-вид

```

<owl:Class rdf:ID="Кластер"/>
<owl:Class rdf:ID="Экранированная">
    <rdfs:subClassOf>
        <owl:Class rdf:about="#Витая_пара"/>
    </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Оптоволокно">
    <rdfs:subClassOf rdf:resource="#Кабель"/>
</owl:Class>

```

/// Инициализация отношений DataTypeProperties онтологии

```

<owl:DatatypeProperty rdf:ID="имеетОписание">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>

```

```

    <rdfs:domain rdf:resource="#СКС"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="имеетВнутреннююПропускнуюСпособность">
    <rdfs:domain rdf:resource="#Коммутатор"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>

```

/// Инициализация отношений ObjectProperties онтологии

```

<owl:ObjectProperty rdf:ID="включаетПоказатель">
    <rdfs:domain rdf:resource="#Ситуация"/>
    <rdfs:range rdf:resource="#Параметры_работоспособности_сети"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="имеетТопологию">
    <rdfs:domain rdf:resource="#СКС"/>
    <rdfs:range rdf:resource="#Топология_сети"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="имеетТенденцию">
    <rdfs:range rdf:resource="#Тенденция"/>
    <rdfs:domain rdf:resource="#Параметры_работоспособности_сети"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="содержитХарактеристику">
    <rdfs:domain rdf:resource="#Ситуация"/>
    <rdfs:range>
    <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
            <owl:Class rdf:about="#Аппаратное_обеспечение"/>
            <owl:Class rdf:about="#Технология_построения_сети"/>
            <owl:Class rdf:about="#Топология_сети"/>
            <owl:Class rdf:about="#Уровень_передачи_данных"/>
            <owl:Class rdf:about="#СКС"/>
        </owl:unionOf>
    </owl:Class>

```

```

    <owl:Class rdf:about="#Программное_обеспечение"/>
  </owl:unionOf>
</owl:Class>
</rdfs:range>
</owl:ObjectProperty>

```

/// Определение свойств отношений онтологии

```

<FunctionalObjectProperty>
  <ObjectProperty IRI="#имеетХарактер"/>
</FunctionalObjectProperty>
<TransitiveObjectProperty>
  <ObjectProperty IRI="#имеетХарактер"/>
</TransitiveObjectProperty>

```

/// Инициализация взвешенных классов онтологии

```

<AnnotationAssertion>
  <AnnotationProperty IRI="#fuzzyLabel"/>
  <IRI>#Рouters</IRI>
  <Literal datatypeIRI="&rdf;PlainLiteral"><fuzzyOwl2 fuzzyType="concept">
<Concept type="weighted" value="1.0" base="Маршрутизатор" />;
</fuzzyOwl2></Literal>
</AnnotationAssertion>

```

```

<AnnotationAssertion>
  <AnnotationProperty IRI="#fuzzyLabel"/>
  <IRI>#ПО_мониторинга_и_анализа_сетей</IRI>
  <Literal datatypeIRI="&rdf;PlainLiteral"><fuzzyOwl2 fuzzyType="concept">
<Concept type="weighted" value="0.7" base="Экспертные_системы" />
</fuzzyOwl2></Literal>
</AnnotationAssertion>

```

```

    /// Инициализация лингвистических значений и функций принадлежности
    <AnnotationAssertion>
    <AnnotationProperty IRI="#fuzzyLabel"/>
    <IRI>#ВысокаяНагрузка</IRI>
    <Literal datatypeIRI="&rdf;PlainLiteral">fuzzyOwl2 fuzzyType="datatype";
Datatype type="rightshoulder"; a="15.0"; b="30.0";/fuzzyOwl2</Literal>
    </AnnotationAssertion>

```

```

<AnnotationAssertion>
    <AnnotationProperty IRI="#fuzzyLabel"/>
    <IRI>#МалоеЧислоКоллизий</IRI>
    <Literal datatypeIRI="&rdf;PlainLiteral"><fuzzyOwl2 fuzzyType="datatype">
<Datatype type="leftshoulder" a="3.0" b="5.4" />
</fuzzyOwl2></Literal>
    </AnnotationAssertion>

```

```

    /// Инициализация нечетких модификаторов
    <AnnotationAssertion>
    <AnnotationProperty IRI="#fuzzyLabel"/>
    <IRI>#Очень</IRI>
    <Literal datatypeIRI="&rdf;PlainLiteral"><fuzzyOwl2 fuzzyType="modifier">
<Modifier type="linear" c="0.9" />
</fuzzyOwl2></Literal>
    </AnnotationAssertion>

```

```

    <AnnotationAssertion>
    <AnnotationProperty IRI="#fuzzyLabel"/>
    <IRI>#Слабый</IRI>
    <Literal datatypeIRI="&rdf;PlainLiteral"><fuzzyOwl2 fuzzyType="modifier">

```



```
<Modifier type="linear" c="0.4" />  
</fuzzyOwl2></Literal>  
  </AnnotationAssertion>
```

ПРИЛОЖЕНИЕ Ж. КОД КЛАССА РАБОТЫ С НЕЧЁТКИМИ ОБЪЕКТАМИ OWL-ОНТОЛОГИИ

```
public FuzzyOWLAPIReasoner() {
    public FuzzyOWLAPIReasoner(Set<KnowledgeSource> sources) {
        super(sources);
    }

    public static Collection<ConfigOption<?>> createConfigOptions() {
        Collection<ConfigOption<?>> options = new
        LinkedList<ConfigOption<?>>();
        StringConfigOption type = new StringConfigOption("reasonerType",
        "FaCT++, HermiT, OWLlink or Pellet, which means \"fact\", \"hermit\",
        \"owlLink\" or \"pellet\"\", \"pellet\", false, true);
        type.setAllowedValues(new String[] { "fact", "hermit", "owlLink", "pellet",
        "fuzzydl" });

        try {
            URLConfigOption owlLinkURL = new
            URLConfigOption("owlLinkURL", "the URL to the remote OWLlink server",
            new URL("http://localhost:8080/"), false, true);
            options.add(owlLinkURL);
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
        return options;
    }

    @Override
    public <T> void applyConfigEntry(ConfigEntry<T> entry) throws
    InvalidConfigOptionValueException {
```

```

    }
    @Override
    public void init() throws ComponentInitException {
        atomicConcepts = new TreeSet<NamedClass>(conceptComparator);
        atomicRoles = new TreeSet<ObjectProperty>(roleComparator);
        datatypeProperties = new TreeSet<DatatypeProperty>();
        individuals = new TreeSet<Individual>();
        prefixes.putAll(((PrefixOWLOntologyFormat)
            format).getPrefixName2PrefixMap());
        baseURI = ((PrefixOWLOntologyFormat)
            format).getDefaultPrefix();
        prefixes.remove("");
    }
    } else {
        KB kb = ((AbstractKnowledgeSource)source).toKB();
        // System.out.println(kb.toString(null,null));

        IRI ontologyURI = IRI.create("http://example.com");
        ontology = null;
        try {
            ontology = manager.createOntology(ontologyURI);
        } catch (OWLOntologyCreationException e) {
            e.printStackTrace();
        }
        OWLAPIAxiomConvertVisitor.fillOWLOntology(manager,
ontology, kb);

        owlAPIOntologies.add(ontology);
        allImports.add(ontology);
        atomicConcepts.addAll(kb.findAllAtomicConcepts());
        atomicRoles.addAll(kb.findAllAtomicRoles());
    }
}

```

```

        individuals.addAll(kb.findAllIndividuals());
    }
}

try {
    ontology = manager.createOntology(IRI.create("http://dl-
learner/all"), new HashSet<OWLOntology>(owlAPIOntologies));
} catch (OWLOntologyCreationException e1) {
    e1.printStackTrace();
}

ReasonerProgressMonitor progressMonitor = new
NullReasonerProgressMonitor();

FreshEntityPolicy freshEntityPolicy = FreshEntityPolicy.ALLOW;
long timeOut = Integer.MAX_VALUE;
IndividualNodeSetPolicy individualNodeSetPolicy =
IndividualNodeSetPolicy.BY_NAME;

OWLReasonerConfiguration conf = new
SimpleConfiguration(progressMonitor, freshEntityPolicy, timeOut,
individualNodeSetPolicy);

if(reasoner == null) {
    try {
        reasoner = new
FuzzyDLReasonerManager(((OWLFile)sources.iterator().next()).getURL().toStri
ng(), ontology, conf, factory, baseURI);
    } catch (Exception e) {
        manager = OWLManager.createOWLOntologyManager();
        Comparator<OWLNamedObject> namedObjectComparator = new
Comparator<OWLNamedObject>() {
            public int compare(OWLNamedObject o1, OWLNamedObject o2)

```

```

{
    return o1.getIRI().compareTo(o2.getIRI());
}

};

Set<OWLClass> classes = new
TreeSet<OWLClass>(namedObjectComparator);

Set<OWLObjectProperty> owlObjectProperties = new
TreeSet<OWLObjectProperty>(namedObjectComparator);

Set<OWLObjectProperty> owlDatatypeProperties = new
TreeSet<OWLObjectProperty>(namedObjectComparator);

Set<OWLNamedIndividual> owlIndividuals = new
TreeSet<OWLNamedIndividual>(namedObjectComparator);

Set<OWLOntology> allImports = new HashSet<OWLOntology>();
prefixes = new TreeMap<String,String>();

for(KnowledgeSource source : sources) {

    if (source instanceof OWLOntologyKnowledgeSource) {
        ontology = ((OWLOntologyKnowledgeSource)
source).createOWLOntology(manager);
        owlAPIOntologies.add(ontology);
    }

    if(source instanceof OWLFile || source instanceof
SparqlKnowledgeSource || source instanceof OWLAPIOntology) {

        Set<OWLOntology> imports = manager.getImportsClosure(ontology);
        allImports.addAll(imports);
        System.out.println(imports);
    }
}

```

```

        for (OWLOntology ont : imports) {
            classes.addAll(ont.getClassesInSignature());
            owlObjectProperties.addAll(ont.getObjectPropertiesInSignature());
            owlDatatypeProperties.addAll(ont.getDataPropertiesInSignature());
            owlIndividuals.addAll(ont.getIndividualsInSignature());
        }

        OWLOntologyFormat format =
manager.getOntologyFormat(ontology);
        if (format instanceof PrefixOWLOntologyFormat) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    } else {
        try {
            OWLlinkHTTPXMLReasonerFactory factory = new
OWLlinkHTTPXMLReasonerFactory();
            URL url = owlLinkURL;
            OWLlinkReasonerConfiguration config = new
OWLlinkReasonerConfiguration(url);
            reasoner = factory.createNonBufferingReasoner(ontology,
config);

            System.out.println(reasoner.getReasonerName());
        } catch (Exception e) {
            //
            e.printStackTrace();
            throw new ComponentInitException(e);
        }
    }
}

```